DA16200/DA16600 Getting Started with AWS IoT Core

The DA16200/DA16600 is a highly integrated ultra-low power Wi-Fi system on chip (SoC) that allows you to develop a complete Wi-Fi solution on a single chip. This document is a DA16200/DA16600 manual intended to help new or existing developers quickly get started using AWS IoT Core.

Contents

Figures

Tables

1. Terms and Definitions

AP	Access Point
API	Application Programming Interface
AWS	Amazon Web Services
DEVKT	Development Kits
DPM	Dynamic Power Management
DTIM	Delivery Traffic Indication Map
IDE	Integrated Development Environment
IoT	Internet of Things
LE	Low Energy
MCU	Micro-Controller Unit
OTA	Over the Air
SDK	Software Development Kit
TIM	Traffic Indication Map

2. References

[1] DA16200MOD, Datasheet, Renesas Electronics.
(DA16200MOD - Ultra-Low Power Wi-Fi Modules for Battery Powered IoT Devices | Renesas)

[2] DA16600MOD, Datasheet, Renesas Electronics.
(DA16600MOD - Ultra-Low Power Wi-Fi + Bluetooth® Low Energy Combo Modules for Battery Powered IoT Devices | Renesas)

[3] UM-WI-056, DA16200 DA16600 FreeRTOS Getting Started Guide, User Manual, Renesas Electronics.
(UM-WI-056 DA16200 DA16600 FreeRTOS Getting Started Guide (renesas.com))

[4] UM-WI-042, DA16200 DA16600 Provisioning Mobile App for Android/iOS, User Manual, Renesas Electronics.
(DA16200 DA16600 Provisioning Mobile App (renesas.com))

Note 1 References are for the latest published version, unless otherwise indicated.

3. Overview

The DA16200 Wi-Fi Development Kit (DEVKIT) provides a quick and easy method to start developing battery powered applications and products using ultra-low power Wi-Fi. The ultra-low power DA16200 chipset is the world's lowest power Wi-Fi device specifically designed to meet the requirements for power sensitive wireless applications.

The DA16600MOD-DEVKT provides a host board based on the DA16600 Wi-Fi + Bluetooth® Low Energy (LE) module supporting a USB connection to a personal computer for evaluation and development of low power Wi-Fi and Bluetooth® LE applications.

3.1 Benefits

- Ultra-low power Wi-Fi technology
- More than 1-year battery life for most applications
- Industry leading wireless range
- Fully integrated Wi-Fi system on chip (SoC)
- Comprehensive security capabilities
- Easy to use development tools means shorter time to market.

3.2 Features

- Highly integrated ultra-low power DA16200 Wi-Fi system module
- Best Radio Frequency performance
- SoC runs full networking OS and TCP/IP stack
- Built-in 4-channel auxiliary ADC for sensor interfaces
- Built-in hardware crypto engines for advanced security features
- Complete software stack
- eMMC/SD expanded memory.

3.3 Applications

DA16200MOD/DA16600MOD is a full offload SoC for IoT applications, such as:

- Security systems
- Door locks
- Thermostats
- Garage door openers
- Blinds
- Lighting control
- Sprinkler systems
- Video camera security systems
- Smart appliances
- Video doorbell.

4. Hardware Description

4.1 Datasheet

4.1.1 DA16200MOD

The DA16200MOD is a fully integrated Wi-Fi® module with ultra-low power consumption, best RF performance, and easy development environment. For more information, see Ref. [1].

4.1.2 DA16600MOD

The DA16600 modules provide a convenient way to add both low power Wi-Fi and low power Bluetooth® LE functionality to your device. For more information, see Ref. [2].

NOTE
To purchase the development kits, click the following links: ▪ DA16200MOD-DEVKT Renesas Electronics Corporation | Development Boards, Kits, Programmers | DigiKey ▪ DA16600MOD-DEVKT Renesas Electronics Corporation | Development Boards, Kits, Programmers | DigiKey

5. Set Up Development Environment

You can develop Wi-Fi applications for the DA16200 using the DA16200 FreeRTOS Software Development Kit (SDK) and the Renesas e^{2}studio IDE on either a Windows 10 or Linux based development system.

To set up SDK, complete the following steps:

- Install and configure the e^{2}studio IDE
- Import the DA16200 SDK into the e^{2}studio, and build an application
- Setting up the evaluation board
- Download and test the application
- Use J-Link debugger to debug the application.

For more information, see DA16200 DA16600 FreeRTOS Getting Started Guide, Ref. [3].

6. AWS IoT

The DA16200MOD/DA16600MOD is a full offload SoC for IoT applications such as security systems, door locks, and smart applications. This section provides procedures on how to configure AWS IoT for communicating with DA16200/DA16600 IoT devices.

6.1 Configure AWS IoT

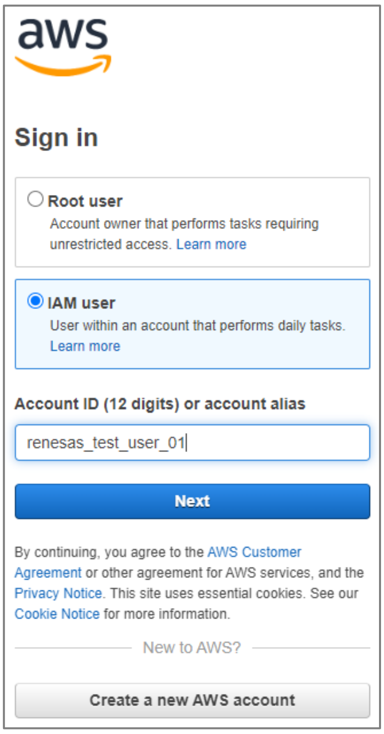
To connect a device to the AWS IoT server, complete the following steps:

1. Sign up AWS account and permissions.
2. Connect devices to AWS IoT.
3. Configure Amazon Cognito user pools and identity pools.
4. Set up Amazon IAM.
5. Create S3 bucket.

6.1.1 Sign Up for AWS Account

To create an AWS account and grant permissions:

1. Go to AWS website and create a free account (https://portal.aws.amazon.com/).
2. Create an administrative user for performing daily administrative tasks.
3. Open the AWS IoT console to get started with AWS IoT.



Sign in

○ Root user
Account owner that performs tasks requiring unrestricted access. Learn more

● IAM user
User within an account that performs daily tasks. Learn more

Account ID (12 digits) or account alias

renesas_test_user_01

Next

By continuing, you agree to the AWS Customer Agreement or other agreement for AWS services, and the Privacy Notice. This site uses essential cookies. See our Cookie Notice for more information.

New to AWS?

Create a new AWS account

Figure 1. Sign up for AWS account

NOTE
If you do not have an AWS account, Renesas Electronics can provide a Thing name that has already been created for testing.

6.1.2 Connect Devices to AWS IoT

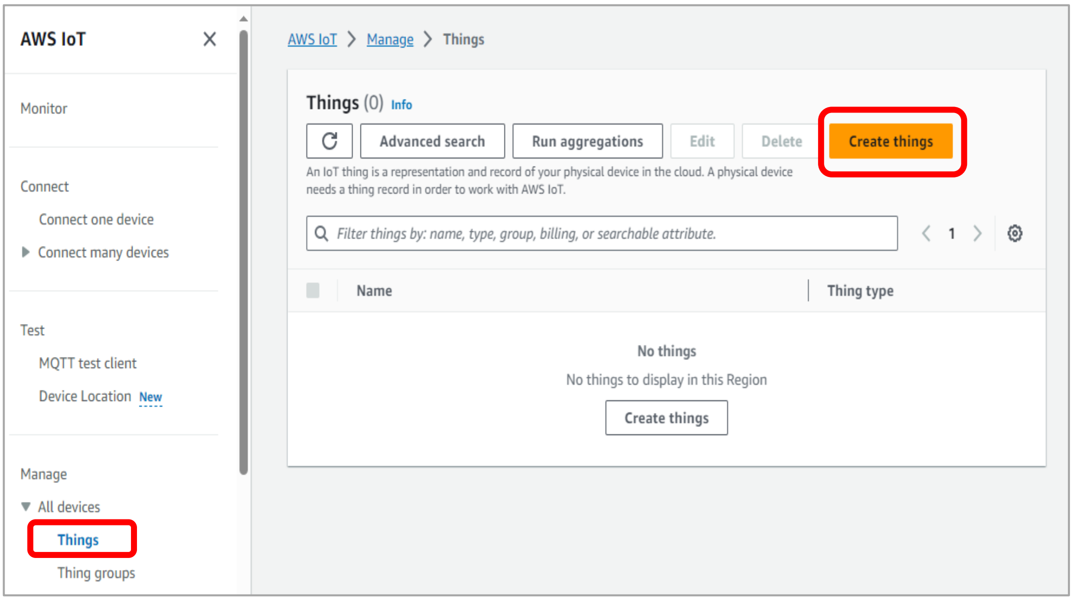
You can configure and manage the thing objects, certificates, rules, jobs, policies, and other elements of IoT solutions through AWS IoT console. Prior to sending data to and receiving data from AWS IoT server, you need to register a device first.

6.1.2.1 Register a Device in Thing Registry

In the Thing Registry, the devices connected to the AWS IoT server are represented by Things. The Thing Registry allows keeping records of all devices that are connected to an AWS IoT account.

To register a device in the Thing Registry:

1. On the AWS IoT console, on the navigation pane, expand **Registry**.
2. Expand **All devices** and click **Things** > **Create things**.

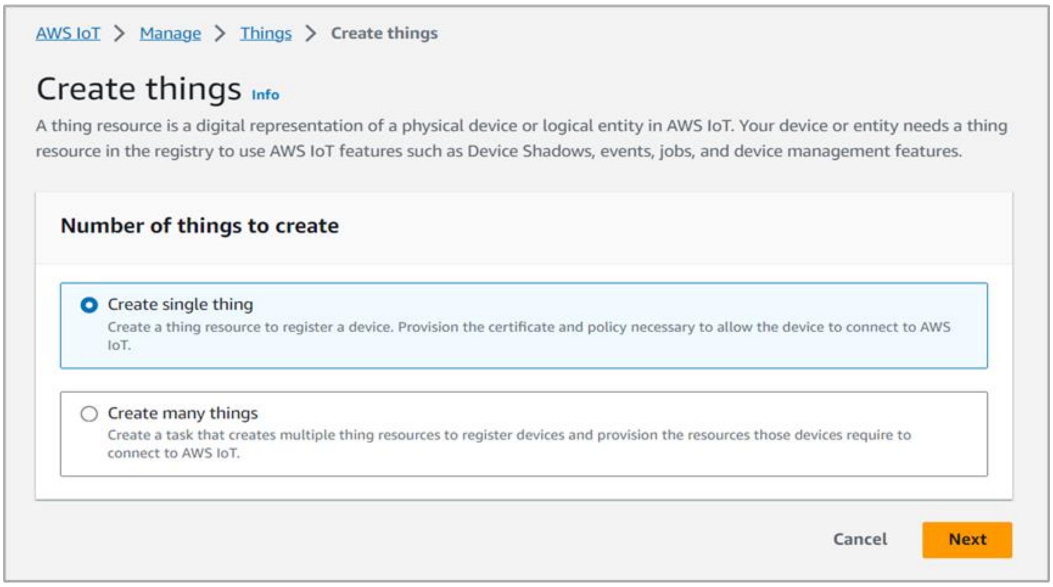


An IoT thing is a representation and record of your physical device in the cloud. A physical device needs a thing record in order to work with AWS IoT.

No things to display in this Region

Figure 2. Register things

3. Select **Create single thing**.



A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

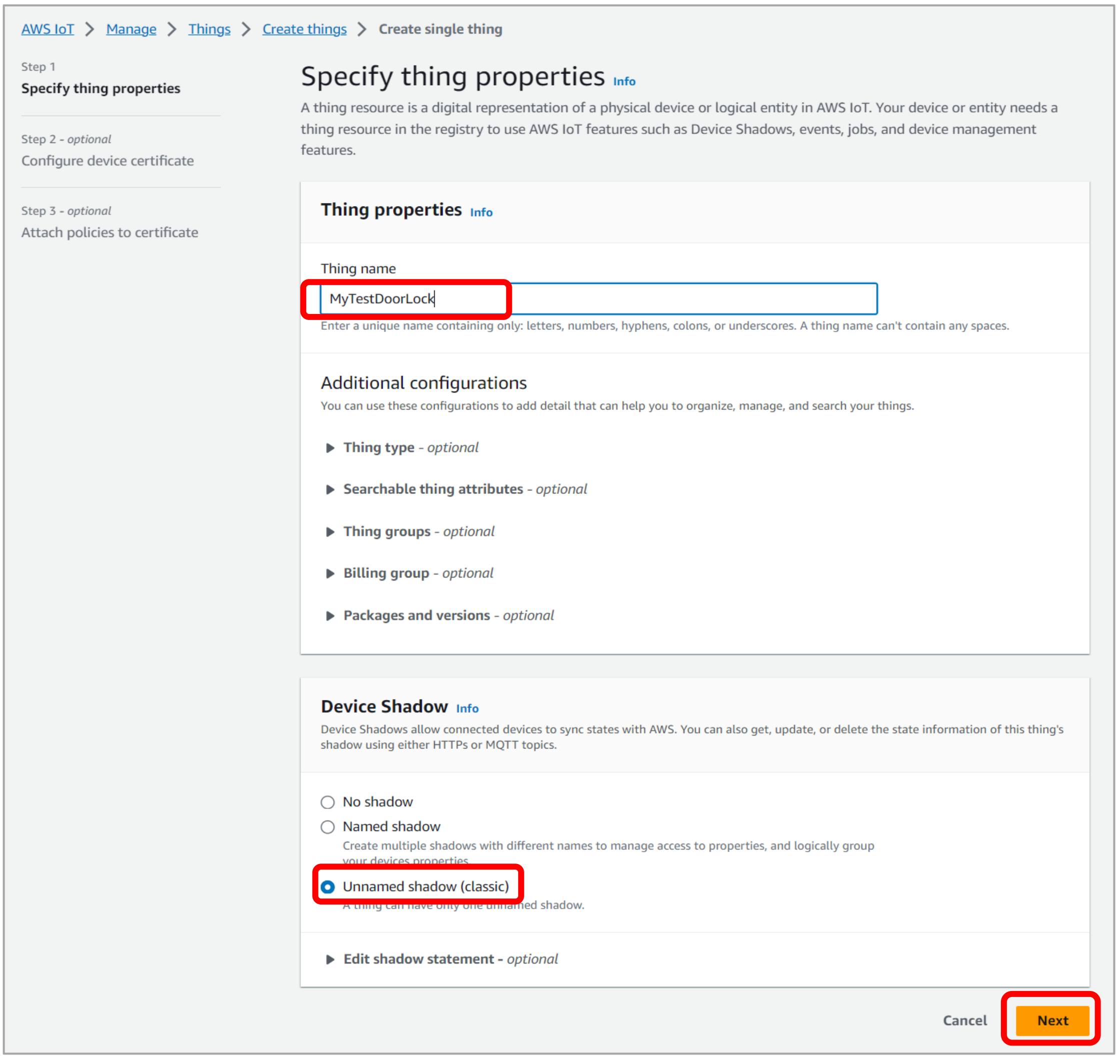
Number of things to create

● Create single thing
Create a thing resource to register a device. Provision the certificate and policy necessary to allow the device to connect to AWS IoT.

○ Create many things
Create a task that creates multiple thing resources to register devices and provision the resources those devices require to connect to AWS IoT.

Figure 3. Create single thing

4. To add the device to the Thing Registry, in the **Thing name** field, enter a device name, for example, "MyTestDoorLock", and under **Device Shadow**, select **Unnamed shadow (classic)** and click **Next**.



AWS IoT > Manage > Things > Create things > Create single thing

Step 1
Specify thing properties

Step 2 - *optional*
Configure device certificate

Step 3 - *optional*
Attach policies to certificate

Specify thing properties Info

A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

Thing properties Info

Thing name
MyTestDoorLock
Enter a unique name containing only: letters, numbers, hyphens, colons, or underscores. A thing name can't contain any spaces.

Additional configurations
You can use these configurations to add detail that can help you to organize, manage, and search your things.

▶ **Thing type** - *optional*
▶ **Searchable thing attributes** - *optional*
▶ **Thing groups** - *optional*
▶ **Billing group** - *optional*
▶ **Packages and versions** - *optional*

Device Shadow Info
Device Shadows allow connected devices to sync states with AWS. You can also get, update, or delete the state information of this thing's shadow using either HTTPs or MQTT topics.

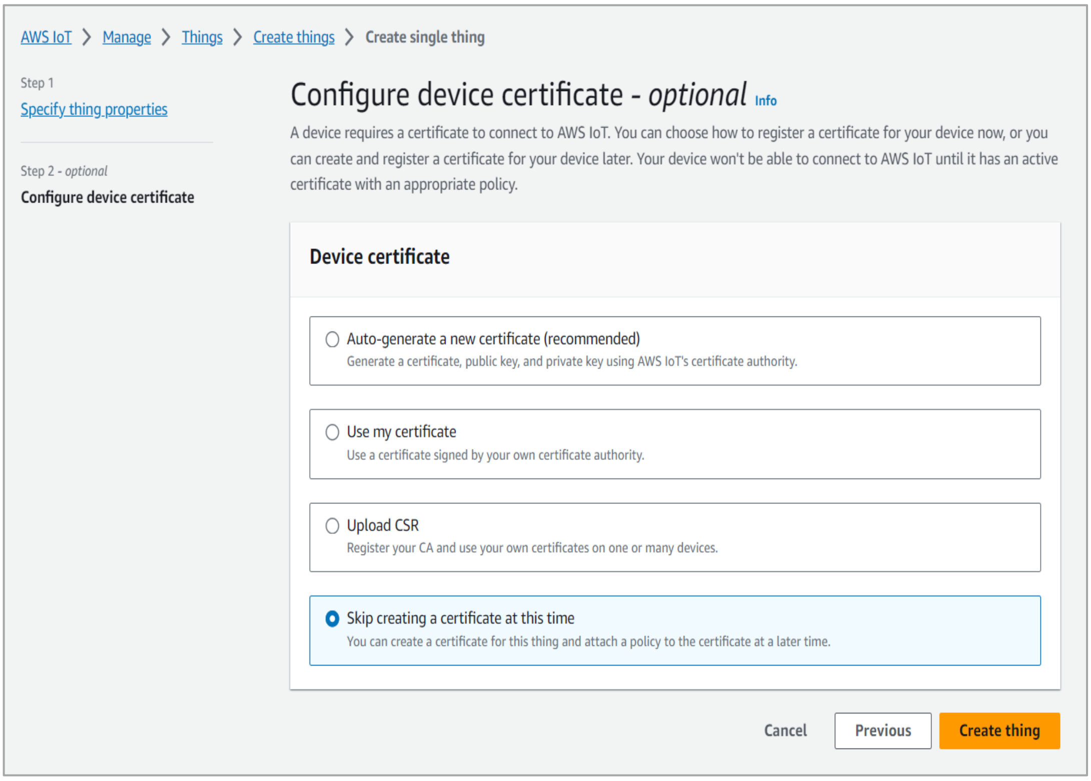
○ No shadow
○ Named shadow
Create multiple shadows with different names to manage access to properties, and logically group your devices properties.
● Unnamed shadow (classic)
A thing can have only one unnamed shadow.

▶ **Edit shadow statement** - *optional*

Cancel Next

Figure 4. Thing name

5. Select **Skip creating a certificate at this time** and click **Create thing**.



AWS IoT > Manage > Things > Create things > Create single thing

Step 1
Specify thing properties

Step 2 - *optional*
Configure device certificate

Configure device certificate - *optional* Info

A device requires a certificate to connect to AWS IoT. You can choose how to register a certificate for your device now, or you can create and register a certificate for your device later. Your device won't be able to connect to AWS IoT until it has an active certificate with an appropriate policy.

Device certificate

○ Auto-generate a new certificate (recommended)
Generate a certificate, public key, and private key using AWS IoT's certificate authority.

○ Use my certificate
Use a certificate signed by your own certificate authority.

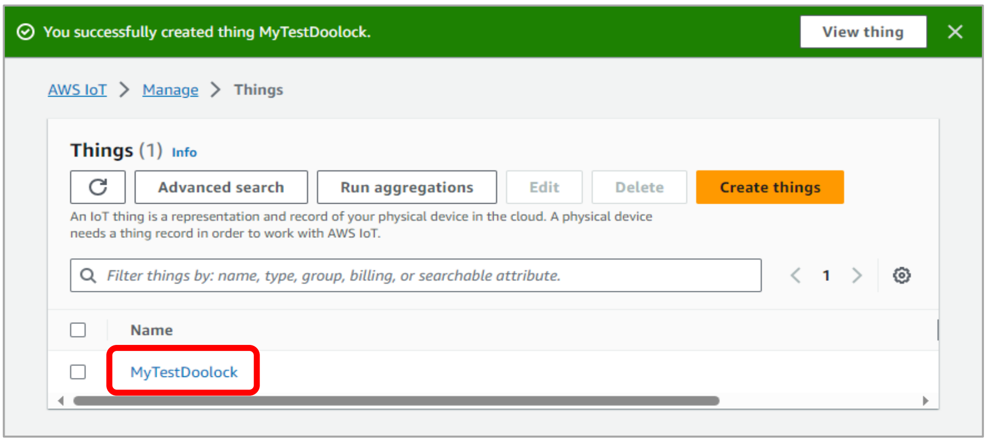
○ Upload CSR
Register your CA and use your own certificates on one or many devices.

● Skip creating a certificate at this time
You can create a certificate for this thing and attach a policy to the certificate at a later time.

Figure 5. Thing without certificate

Now you have the thing created to perform the test and it is named **MyTestDoorLock**.

6. In the **Things** list, click the created thing.



You successfully created thing MyTestDoolock.

AWS IoT > Manage > Things

Things (1) Info

An IoT thing is a representation and record of your physical device in the cloud. A physical device needs a thing record in order to work with AWS IoT.

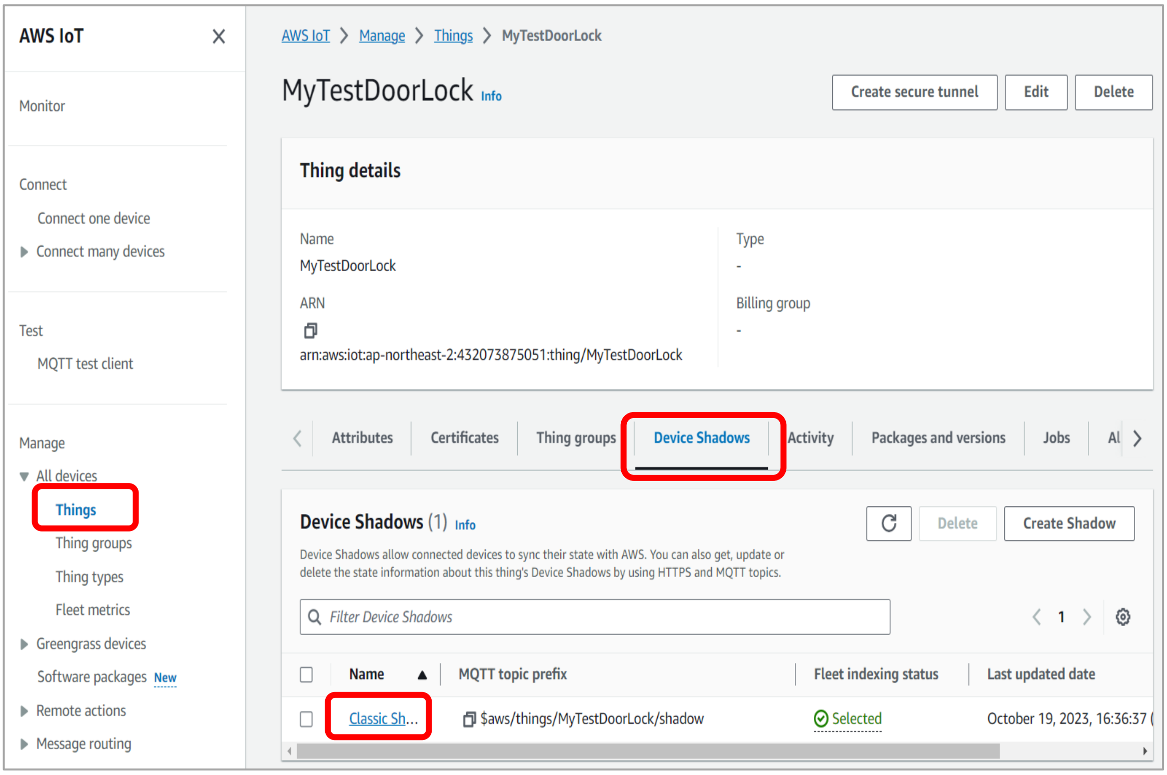
Filter things by: name, type, group, billing, or searchable attribute.

□	Name
□	MyTestDoolock

Figure 6. Created thing

Then, the thing details appear.

7. For the shadow function of the thing, select the **Device Shadows** and click **Classic shadow**.



MyTestDoorLock Info

Thing details

Name
MyTestDoorLock

Type
-

ARN
arn:aws:iot:ap-northeast-2:432073875051:thing/MyTestDoorLock

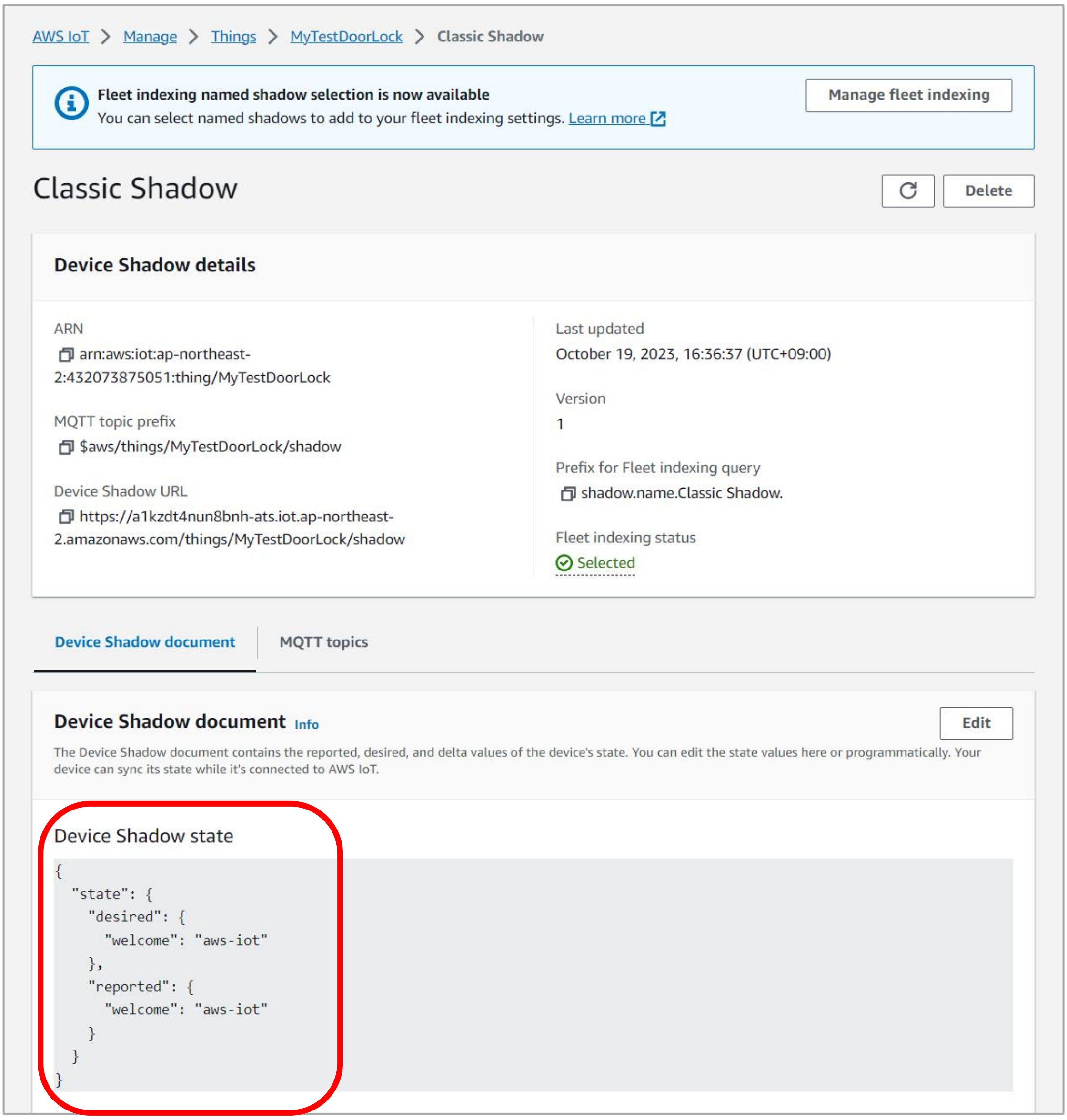
Billing group
-

Device Shadows (1) Info

Device Shadows allow connected devices to sync their state with AWS. You can also get, update or delete the state information about this thing's Device Shadows by using HTTPS and MQTT topics.

Name	MQTT topic prefix	Fleet indexing status	Last updated date
Classic Sh...	$aws/things/MyTestDoorLock/shadow	Selected	October 19, 2023, 16:36:37 (

Figure 7. Classic shadow



AWS IoT > Manage > Things > MyTestDoorLock > Classic Shadow
Fleet indexing named shadow selection is now available
You can select named shadows to add to your fleet indexing settings. Learn more
Manage fleet indexing
Classic Shadow
Delete
Device Shadow details

ARN
arn:aws:iot:ap-northeast-2:432073875051:thing/MyTestDoorLock

MQTT topic prefix
$aws/things/MyTestDoorLock/shadow

Device Shadow URL
https://a1kzdt4nun8bnh-ats.iot.ap-northeast-2.amazonaws.com/things/MyTestDoorLock/shadow

Last updated
October 19, 2023, 16:36:37 (UTC+09:00)

Version
1

Prefix for Fleet indexing query
shadow.name.Classic Shadow.

Fleet indexing status
Selected

The Device Shadow document contains the reported, desired, and delta values of the device's state. You can edit the state values here or programmatically. Your device can sync its state while it's connected to AWS IoT.

Device Shadow state

```
{
  "state": {
    "desired": {
      "welcome": "aws-iot"
    },
    "reported": {
      "welcome": "aws-iot"
    }
  }
}
```

Figure 8. Device shadow document

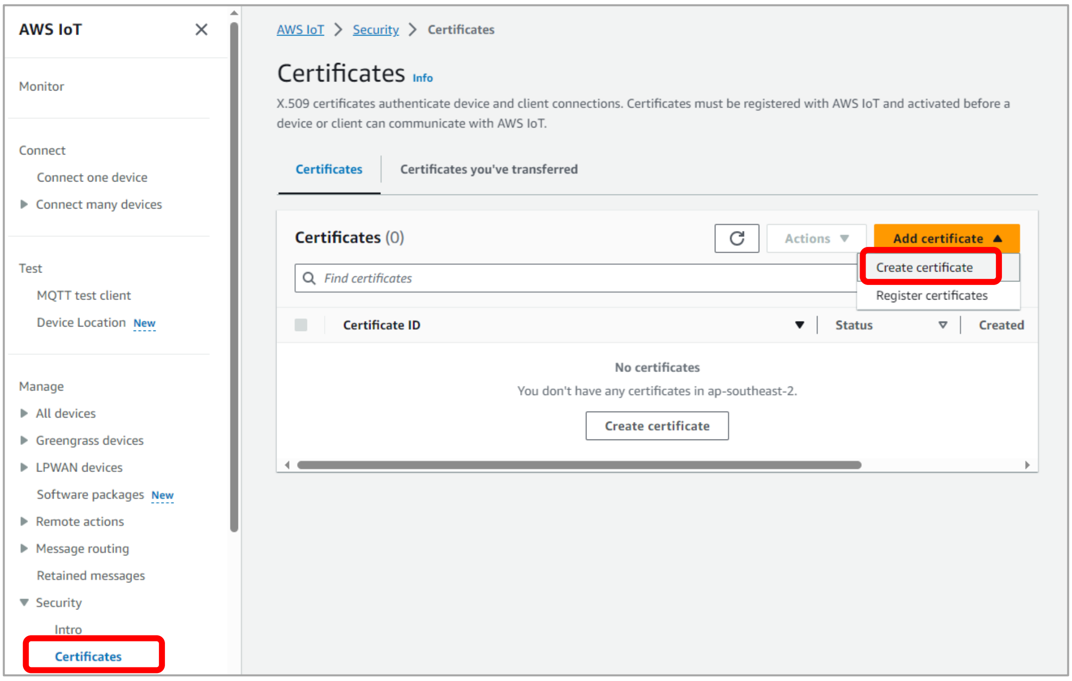
For more information on device shadows for AWS IoT, visit AWS IoT Device Shadow service (https://docs.aws.amazon.com/iot/latest/developerguide/iot-device-shadows.html).

6.1.2.2 Create and Activate Device Certificate

The communication between the device and the AWS IoT is protected by X.509 certificates. You can let the AWS IoT generates a certificate or you can use your own X.509 certificate. This section shows that AWS IoT generates the X.509 certificate.

You should activate the certificates before use. To create and activate a device certificate:

1. On the navigation pane, expand **Security** and click **Certificates,** and then click **Add certificate** > **Create certificate**.

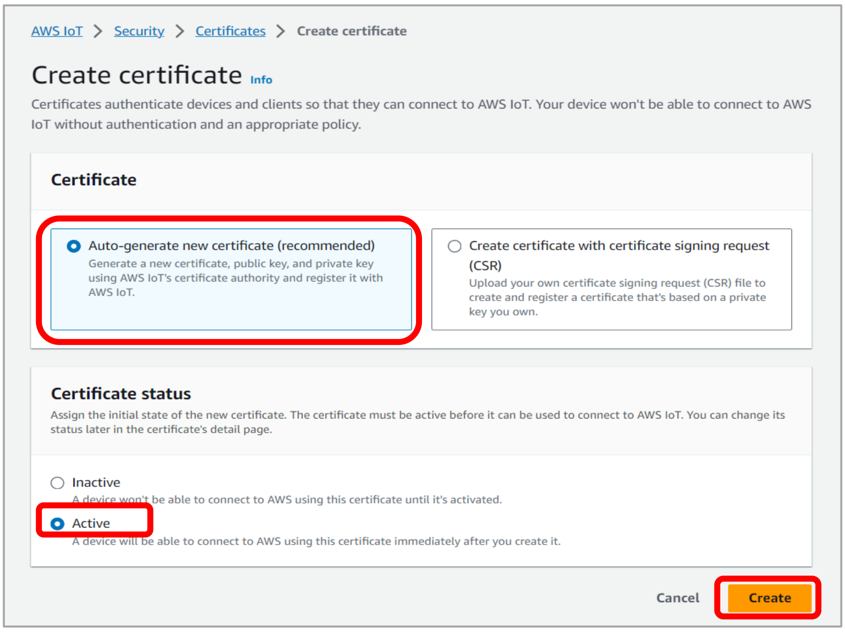


X.509 certificates authenticate device and client connections. Certificates must be registered with AWS IoT and activated before a device or client can communicate with AWS IoT.

You don't have any certificates in ap-southeast-2.

Figure 9. Create certificates

2. Select **Auto-generate new certificate (recommended)** > **Activate** and click **Create**.



Certificates authenticate devices and clients so that they can connect to AWS IoT. Your device won't be able to connect to AWS IoT without authentication and an appropriate policy.

● Auto-generate new certificate (recommended)
Generate a new certificate, public key, and private key using AWS IoT's certificate authority and register it with AWS IoT.

○ Create certificate with certificate signing request (CSR)
Upload your own certificate signing request (CSR) file to create and register a certificate that's based on a private key you own.

Assign the initial state of the new certificate. The certificate must be active before it can be used to connect to AWS IoT. You can change its status later in the certificate's detail page.

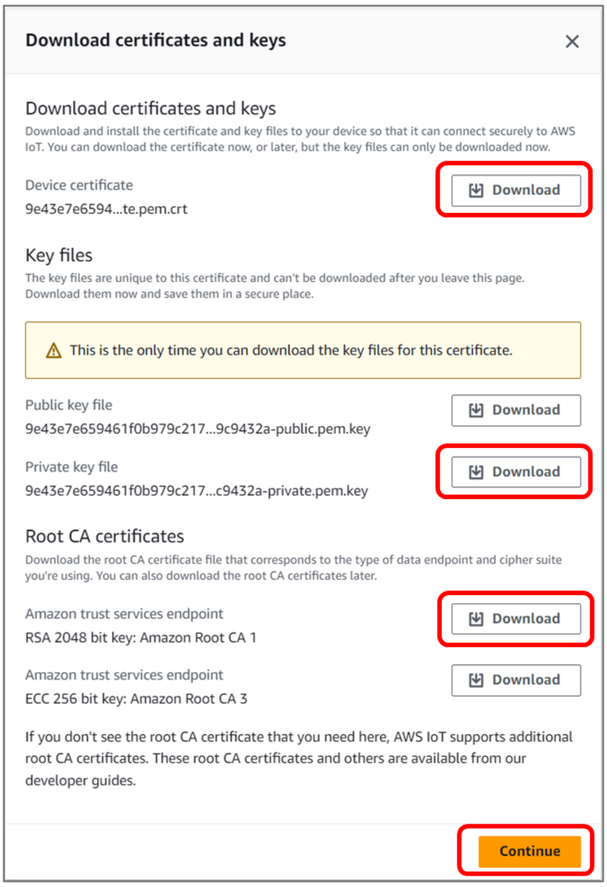
○ Inactive
A device won't be able to connect to AWS using this certificate until it's activated.

● Active
A device will be able to connect to AWS using this certificate immediately after you create it.

Figure 10. Create certificates (continued)

3. There are three required certificates to download.
   On the **Certificate Created** page, download the device certificate, private key, and root CA certificates for AWS IoT, and then save the downloads to your computer, click **Download**.

NOTE
You must save the certificate files before leaving this page. If you leave the page without saving, you no longer have access to the certificate files. Renesas recommends that Device certificate, Private key file, and Root CA should be downloaded in sequential order.



Download certificates and keys ✕

Download certificates and keys

Download and install the certificate and key files to your device so that it can connect securely to AWS IoT. You can download the certificate now, or later, but the key files can only be downloaded now.

Device certificate
9e43e7e6594...te.pem.crt — Download

Key files

The key files are unique to this certificate and can't be downloaded after you leave this page. Download them now and save them in a secure place.

⚠ This is the only time you can download the key files for this certificate.

Public key file
9e43e7e659461f0b979c217...9c9432a-public.pem.key — Download

Private key file
9e43e7e659461f0b979c217...c9432a-private.pem.key — Download

Root CA certificates

Download the root CA certificate file that corresponds to the type of data endpoint and cipher suite you're using. You can also download the root CA certificates later.

Amazon trust services endpoint
RSA 2048 bit key: Amazon Root CA 1 — Download

Amazon trust services endpoint
ECC 256 bit key: Amazon Root CA 3 — Download

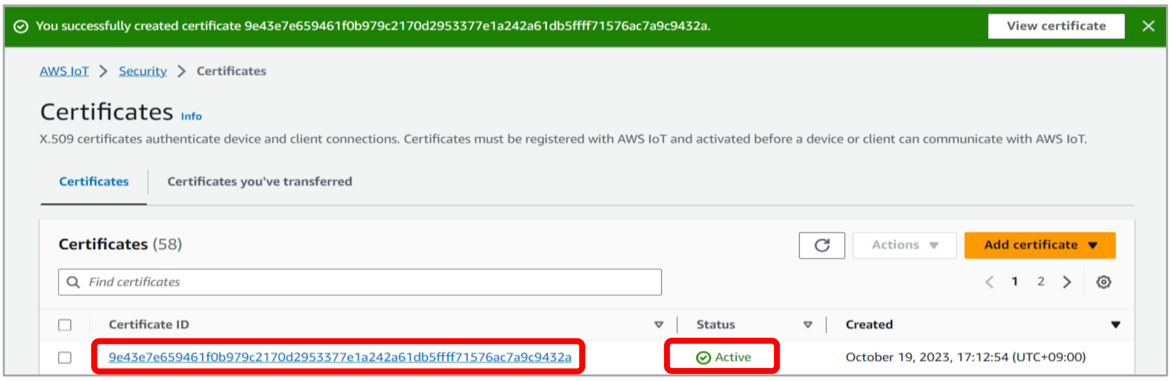
If you don't see the root CA certificate that you need here, AWS IoT supports additional root CA certificates. These root CA certificates and others are available from our developer guides.

Continue

Figure 11. Download certificates and keys

For Root CA, visit the AWS Docs site (https://docs.aws.amazon.com/iot/latest/developerguide/server-authentication.html#server-authentication-certs). Root CA certificates are subjected to expiration and/or revocation.

The certificate status should be **Active** in the list of certificates.



✓ You successfully created certificate 9e43e7e659461f0b979c2170d2953377e1a242a61db5ffff71576ac7a9c9432a. View certificate ✕

AWS IoT > Security > Certificates

Certificates Info

X.509 certificates authenticate device and client connections. Certificates must be registered with AWS IoT and activated before a device or client can communicate with AWS IoT.

Certificates | Certificates you've transferred

Certificates (58) — Actions ▼ — Add certificate ▼

Find certificates — < 1 2 >

□	Certificate ID	Status	Created
□	9e43e7e659461f0b979c2170d2953377e1a242a61db5ffff71576ac7a9c9432a	✓ Active	October 19, 2023, 17:12:54 (UTC+09:00)

Figure 12. Activate certificate

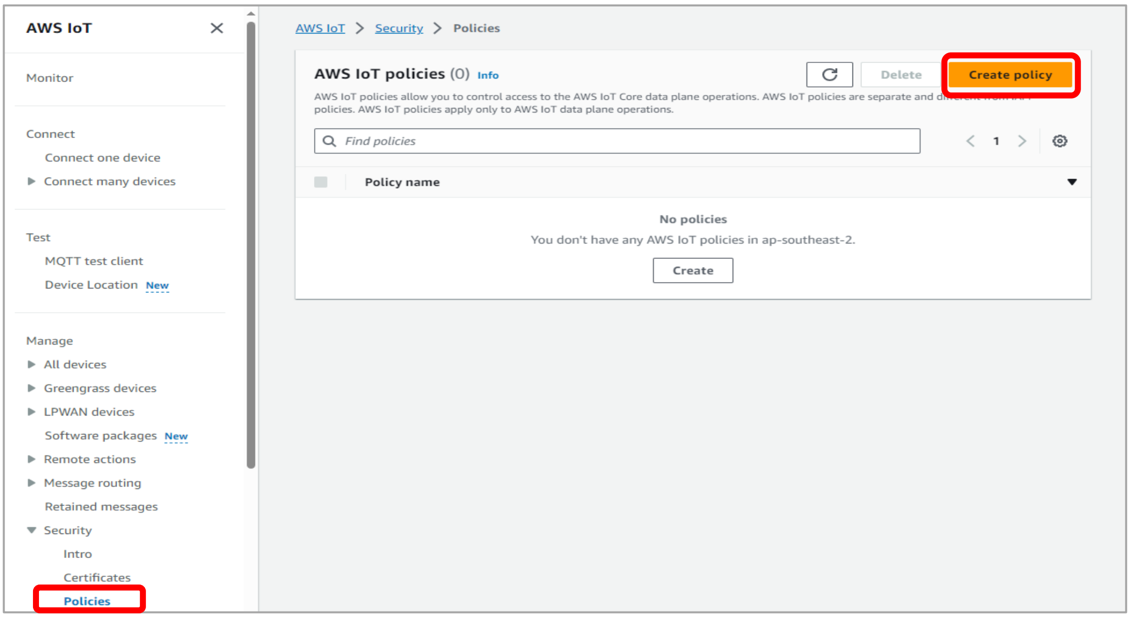
6.1.2.3 Create Policy

The X.509 certificates are used to authenticate the device with the AWS IoT. The AWS IoT policies are used to authorize the device for AWS IoT operations, such as subscribing or publishing to MQTT topics. The device displays its certificate only while connecting to the AWS IoT.

To allow the device for AWS IoT operations, you should create an AWS IoT policy and attach that policy to the device certificate.

To create an AWS IoT policy:

1. On the navigation pane, expand **Security** and click **Policies** > **Create policy**.



AWS IoT policies allow you to control access to the AWS IoT Core data plane operations. AWS IoT policies are separate and different from IAM policies. AWS IoT policies apply only to AWS IoT data plane operations.

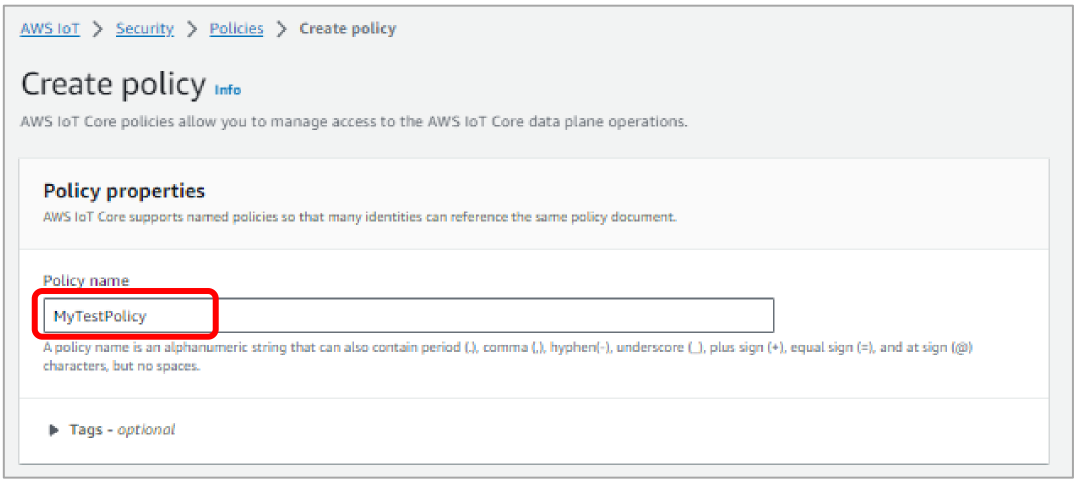
No policies

You don't have any AWS IoT policies in ap-southeast-2.

Figure 13. Create policy

2. On the **Create policy** page:

   a. In the **Policy name** field, under **Policy properties**, enter a name for the policy (for example, MyTestPolicy). Renesas strongly recommends not using personally identifiable information in policy names.



AWS IoT Core policies allow you to manage access to the AWS IoT Core data plane operations.

AWS IoT Core supports named policies so that many identities can reference the same policy document.

Policy name: MyTestPolicy

A policy name is an alphanumeric string that can also contain period (.), comma (,), hyphen(-), underscore (_), plus sign (+), equal sign (=), and at sign (@) characters, but no spaces.

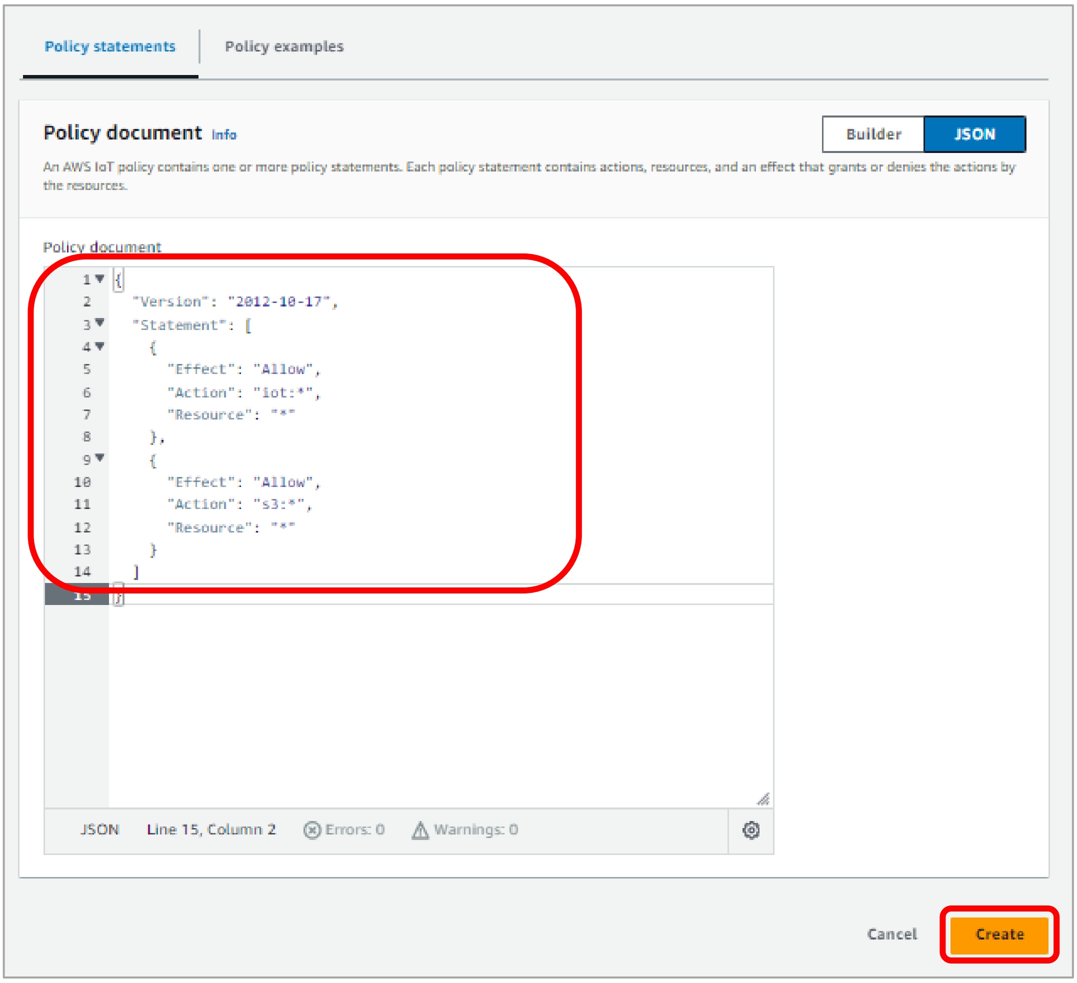
Figure 14. Add policy name

b. Under **Policy document**, select JSON, and then copy and paste the following JSON statement:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": "*"
    }
  ]
}
```

c. After entering the required information, click **Create**.

NOTE
The examples in this document are intended only for development environments. All devices in your production fleet must have credentials with privileges that authorize only intended actions on specific resources. The specific permission policies may vary depending on use cases. Identify the permission policies that best meet the business and security requirements. For more information, see Example Policies and Security Best practices in AWS IoT.



Policy document Info

An AWS IoT policy contains one or more policy statements. Each policy statement contains actions, resources, and an effect that grants or denies the actions by the resources.

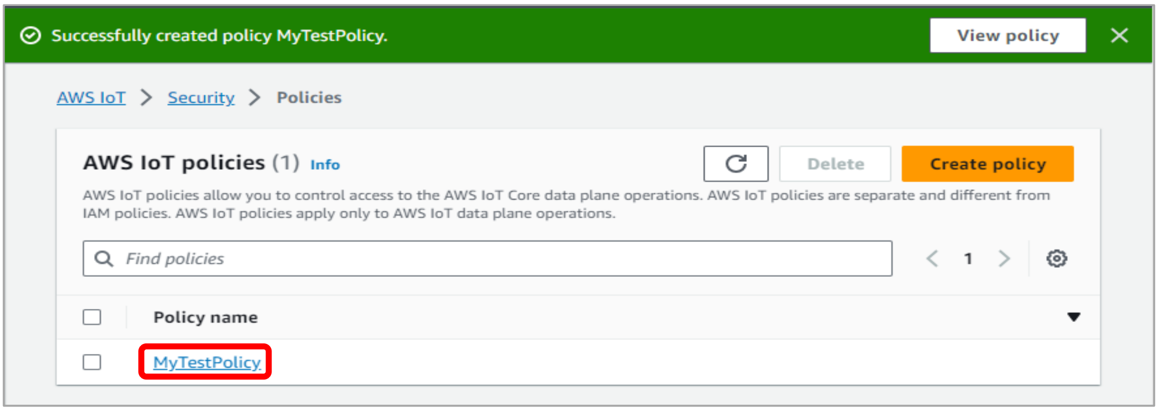
Policy document

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": "*"
    }
  ]
}
```

JSON Line 15, Column 2 Errors: 0 Warnings: 0

Figure 15. Enter JSON policy statement

3. To view the created policies, expand **Security** and click **Policies**.



Successfully created policy MyTestPolicy.

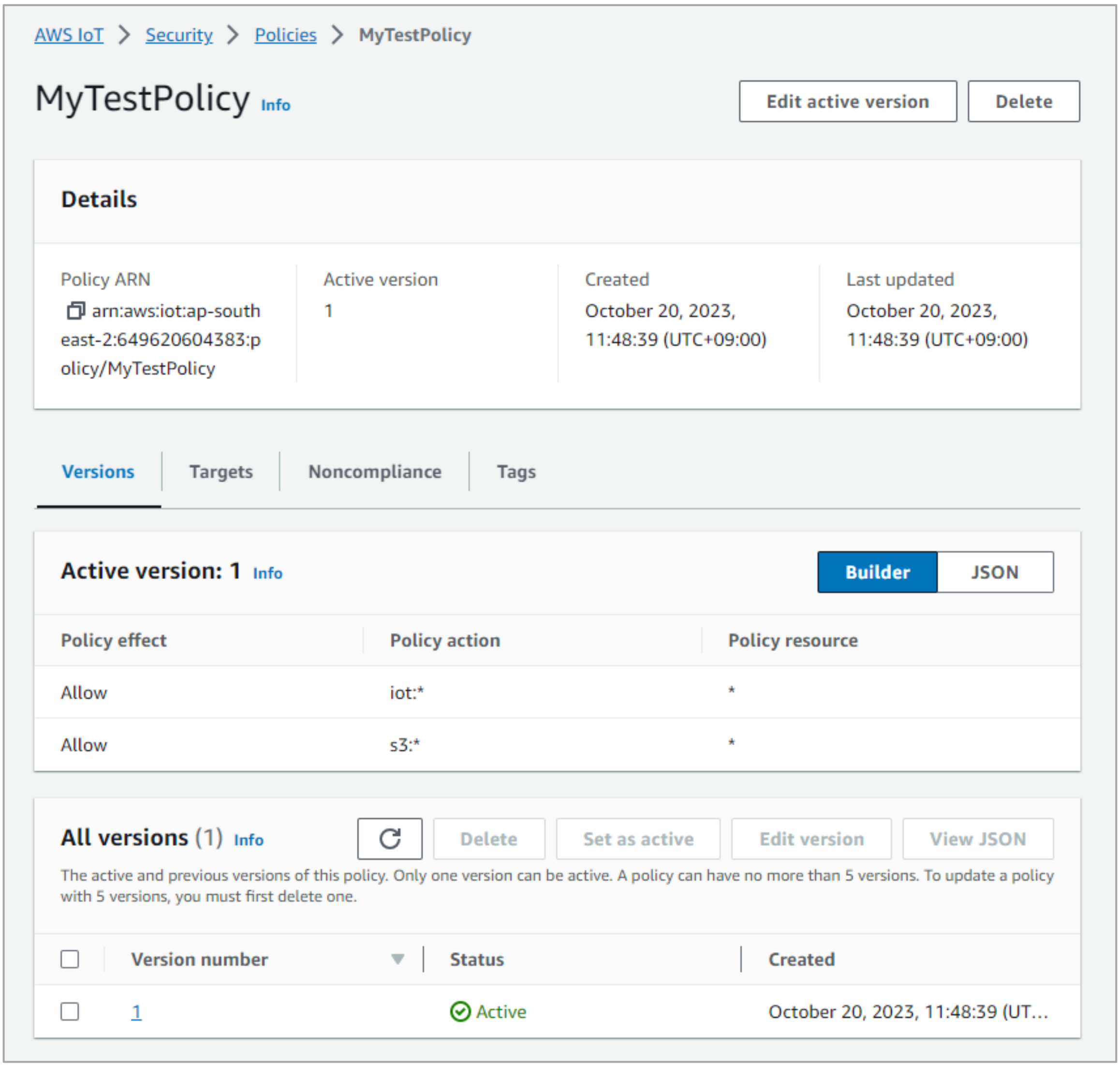
AWS IoT policies (1)

AWS IoT policies allow you to control access to the AWS IoT Core data plane operations. AWS IoT policies are separate and different from IAM policies. AWS IoT policies apply only to AWS IoT data plane operations.

☐	Policy name
☐	MyTestPolicy

Figure 16. Created policy

4. Click the policy to view the details. Figure 17 shows an example of the selected policy content.



MyTestPolicy

Details

Policy ARN	Active version	Created	Last updated
arn:aws:iot:ap-southeast-2:649620604383:policy/MyTestPolicy	1	October 20, 2023, 11:48:39 (UTC+09:00)	October 20, 2023, 11:48:39 (UTC+09:00)

Active version: 1

Policy effect	Policy action	Policy resource
Allow	iot:*	*
Allow	s3:*	*

All versions (1)

The active and previous versions of this policy. Only one version can be active. A policy can have no more than 5 versions. To update a policy with 5 versions, you must first delete one.

☐	Version number	Status	Created
☐	1	✓ Active	October 20, 2023, 11:48:39 (UT...

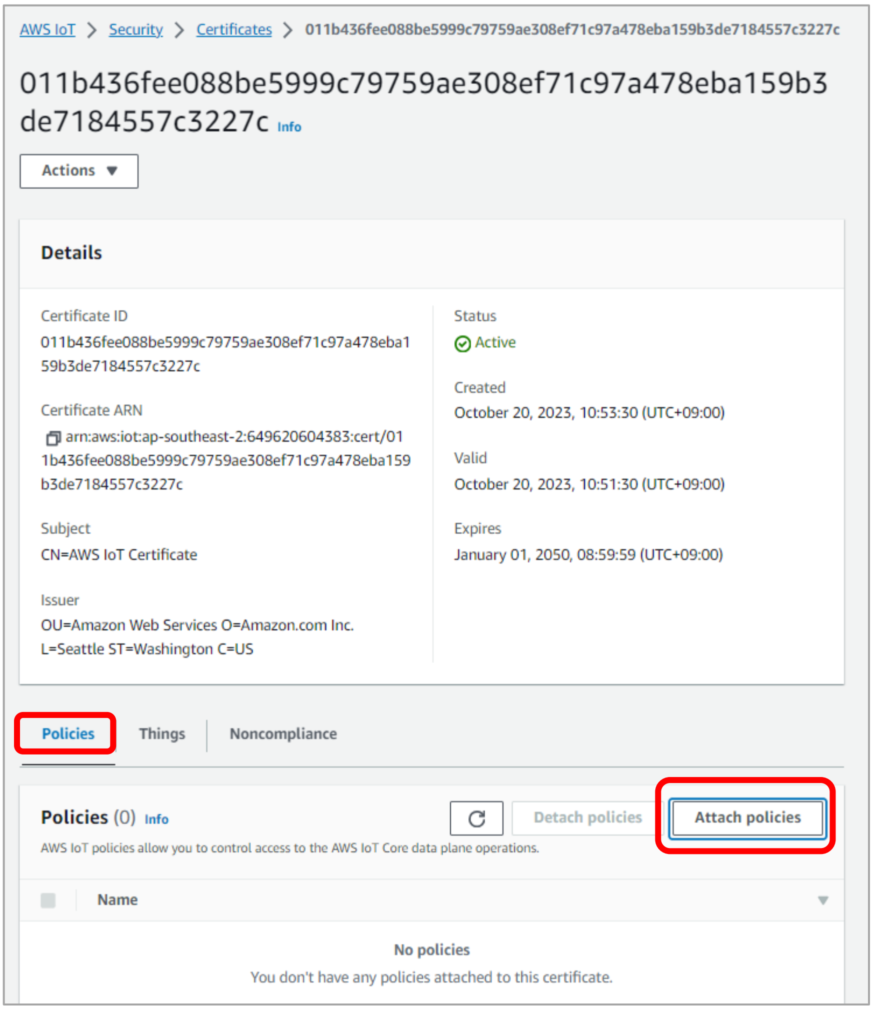
Figure 17. Check created policy

6.1.2.4 Attach Certificate to Thing and Policy

After an AWS IoT policy is created, you must attach that policy to the device certificate. The attachment of an AWS IoT policy to a certificate gives the device the permissions that are specified in the policy.

To attach the AWS IoT policy to a device certificate:

1. Go to the certificate you created, select **Policies** and click **Attach policies**.



AWS IoT > Security > Certificates > 011b436fee088be5999c79759ae308ef71c97a478eba159b3de7184557c3227c

011b436fee088be5999c79759ae308ef71c97a478eba159b3de7184557c3227c Info

Details

Certificate ID
011b436fee088be5999c79759ae308ef71c97a478eba159b3de7184557c3227c

Certificate ARN
arn:aws:iot:ap-southeast-2:649620604383:cert/011b436fee088be5999c79759ae308ef71c97a478eba159b3de7184557c3227c

Subject
CN=AWS IoT Certificate

Issuer
OU=Amazon Web Services O=Amazon.com Inc. L=Seattle ST=Washington C=US

Status
Active

Created
October 20, 2023, 10:53:30 (UTC+09:00)

Valid
October 20, 2023, 10:51:30 (UTC+09:00)

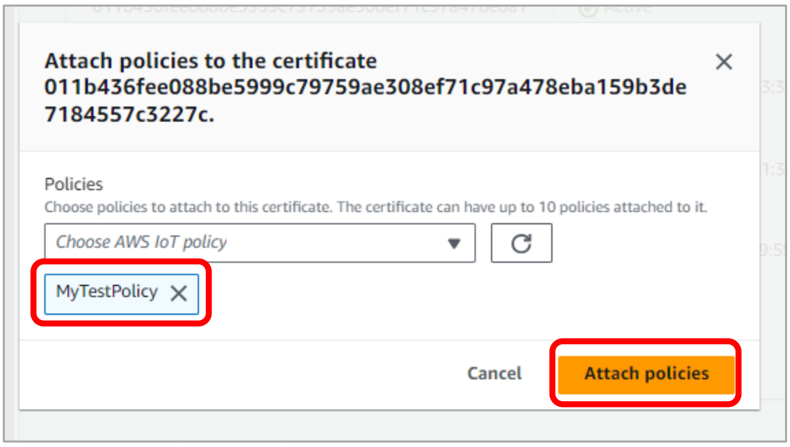
Expires
January 01, 2050, 08:59:59 (UTC+09:00)

Policies (0) Info
AWS IoT policies allow you to control access to the AWS IoT Core data plane operations.

No policies
You don't have any policies attached to this certificate.

Figure 18. Policies

2. Select the created policy and click **Attach policies**.



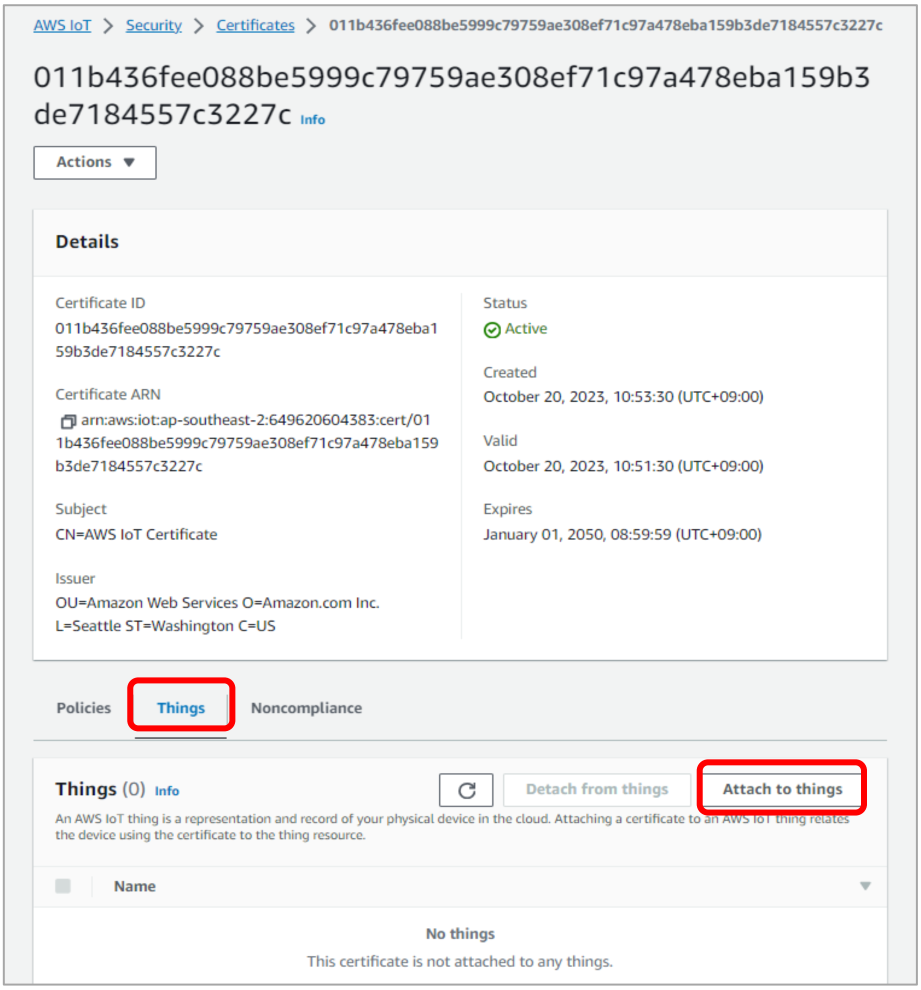
Attach policies to the certificate 011b436fee088be5999c79759ae308ef71c97a478eba159b3de7184557c3227c.

Policies
Choose policies to attach to this certificate. The certificate can have up to 10 policies attached to it.

Figure 19. Attach policy

NOTE
A device should have a certificate, private key, and root CA certificate to authenticate with the AWS IoT. Renesas recommends that you attach the device certificate to the thing that represents the device in AWS IoT. This allows you to create AWS IoT policies that grant permissions based on certificates attached to things.

3. Go to the certificate created by you, select **Things** and click **Attach to things**.



AWS IoT > Security > Certificates > 011b436fee088be5999c79759ae308ef71c97a478eba159b3de7184557c3227c

011b436fee088be5999c79759ae308ef71c97a478eba159b3de7184557c3227c Info

Details

Certificate ID
011b436fee088be5999c79759ae308ef71c97a478eba159b3de7184557c3227c

Certificate ARN
arn:aws:iot:ap-southeast-2:649620604383:cert/011b436fee088be5999c79759ae308ef71c97a478eba159b3de7184557c3227c

Subject
CN=AWS IoT Certificate

Issuer
OU=Amazon Web Services O=Amazon.com Inc. L=Seattle ST=Washington C=US

Status
Active

Created
October 20, 2023, 10:53:30 (UTC+09:00)

Valid
October 20, 2023, 10:51:30 (UTC+09:00)

Expires
January 01, 2050, 08:59:59 (UTC+09:00)

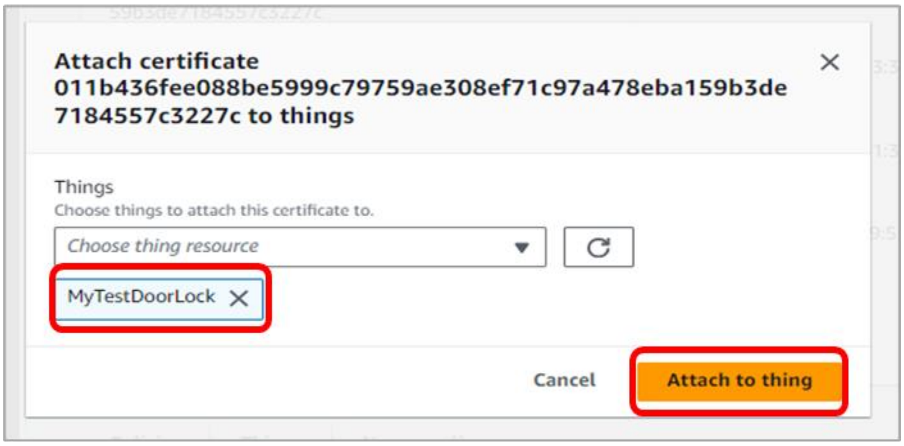
Things (0) Info

An AWS IoT thing is a representation and record of your physical device in the cloud. Attaching a certificate to an AWS IoT thing relates the device using the certificate to the thing resource.

No things
This certificate is not attached to any things.

Figure 20. Attach things to certificate

4. Select the box of the thing that was created and click **Attach to thing**.



Attach certificate 011b436fee088be5999c79759ae308ef71c97a478eba159b3de7184557c3227c to things

Things
Choose things to attach this certificate to.

Choose thing resource

MyTestDoorLock

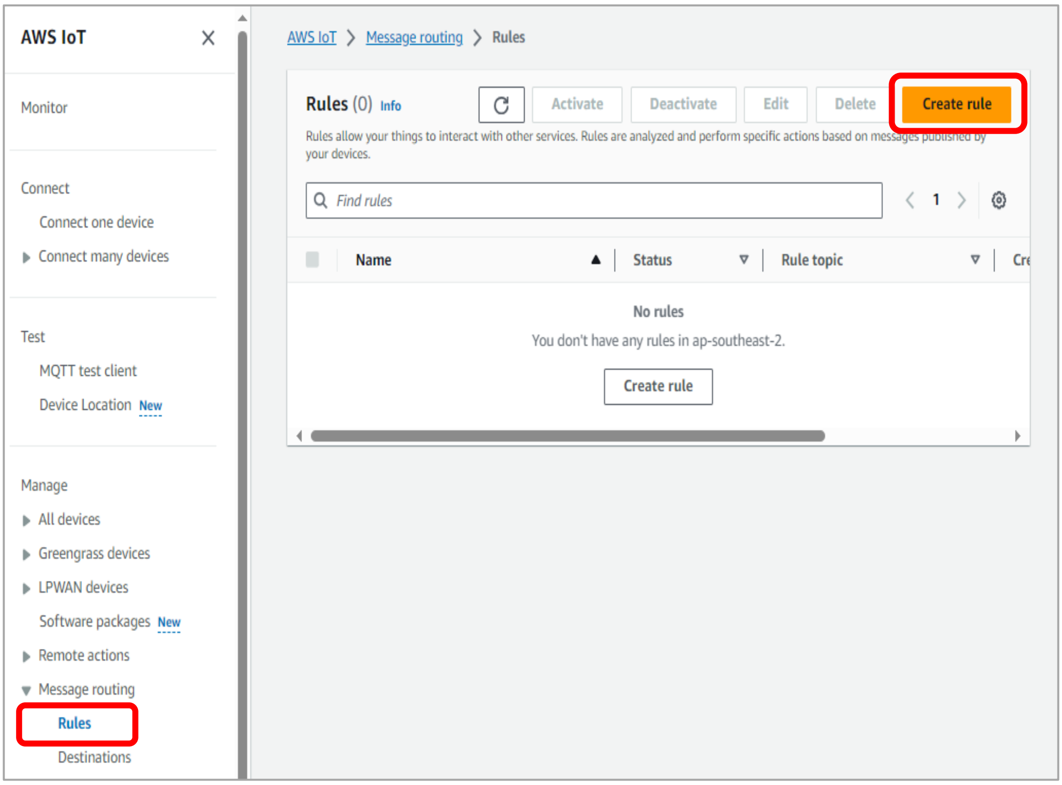
Figure 21. Attach to thing

6.1.2.5 Store Events in S3 Bucket

To store log files for the Door lock:

NOTE
To create Amazon S3 bucket, see Section 6.1.5.

1. Select **AWS console** > **AWS IoT Core**, expand **Message routing** and click **Rules** > **Create rule**.



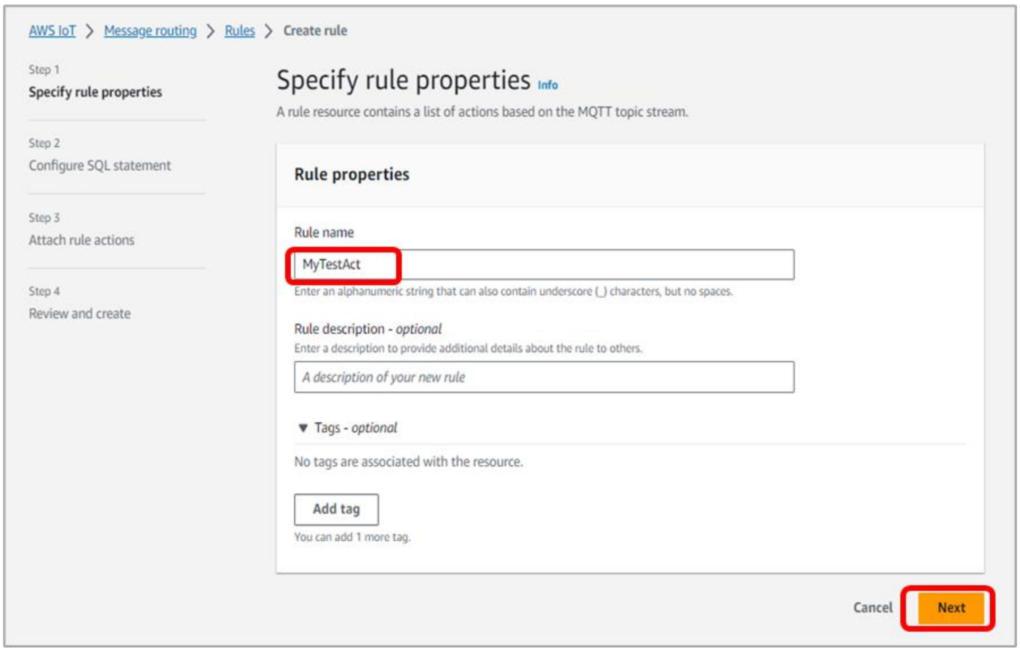
Rules allow your things to interact with other services. Rules are analyzed and perform specific actions based on messages published by your devices.

No rules

You don't have any rules in ap-southeast-2.

Figure 22. Create rule

2. Enter a rule name and click **Next**.



Specify rule properties

A rule resource contains a list of actions based on the MQTT topic stream.

Rule properties

Rule name: MyTestAct

Enter an alphanumeric string that can also contain underscore (_) characters, but no spaces.

Rule description - *optional*

Enter a description to provide additional details about the rule to others.

A description of your new rule

No tags are associated with the resource.

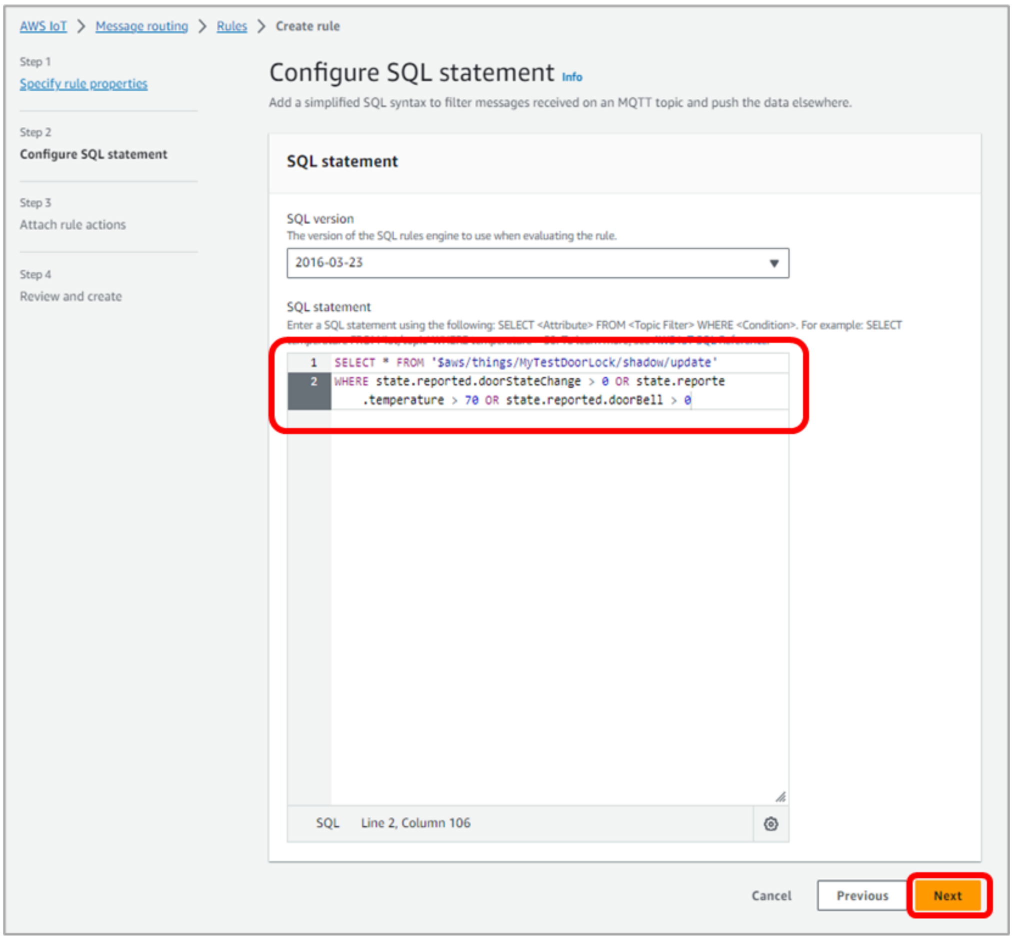
You can add 1 more tag.

Figure 23. Specify rule name

3. Copy and paste the following SQL statement in the SQL statement box and click **Next**.

```
SELECT * FROM '$aws/things/Yourthingname/shadow/update'
WHERE state.reported.doorStateChange > 0 OR state.reported.temperature > 70 OR
state.reported.doorBell > 0
```

Note that the thing name is now **MyTestDoorLock**.



AWS IoT > Message routing > Rules > Create rule

Step 1
Specify rule properties

Step 2
Configure SQL statement

Step 3
Attach rule actions

Step 4
Review and create

Configure SQL statement Info

Add a simplified SQL syntax to filter messages received on an MQTT topic and push the data elsewhere.

SQL statement

SQL version
The version of the SQL rules engine to use when evaluating the rule.
2016-03-23

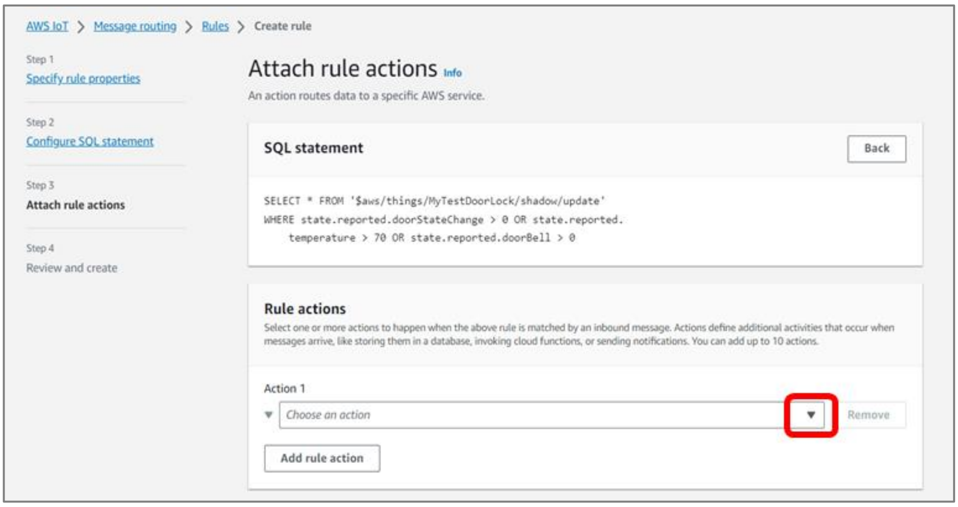
SQL statement
Enter a SQL statement using the following: SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT

```
SELECT * FROM '$aws/things/MyTestDoorLock/shadow/update'
WHERE state.reported.doorStateChange > 0 OR state.reporte
    .temperature > 70 OR state.reported.doorBell > 0
```

SQL   Line 2, Column 106

Figure 24. Configure SQL statement

4. Under **Rule actions**, in the **Action 1** list, select **S3 bucket**.



AWS IoT > Message routing > Rules > Create rule

Step 1
Specify rule properties

Step 2
Configure SQL statement

Step 3
Attach rule actions

Step 4
Review and create

Attach rule actions Info

An action routes data to a specific AWS service.

SQL statement

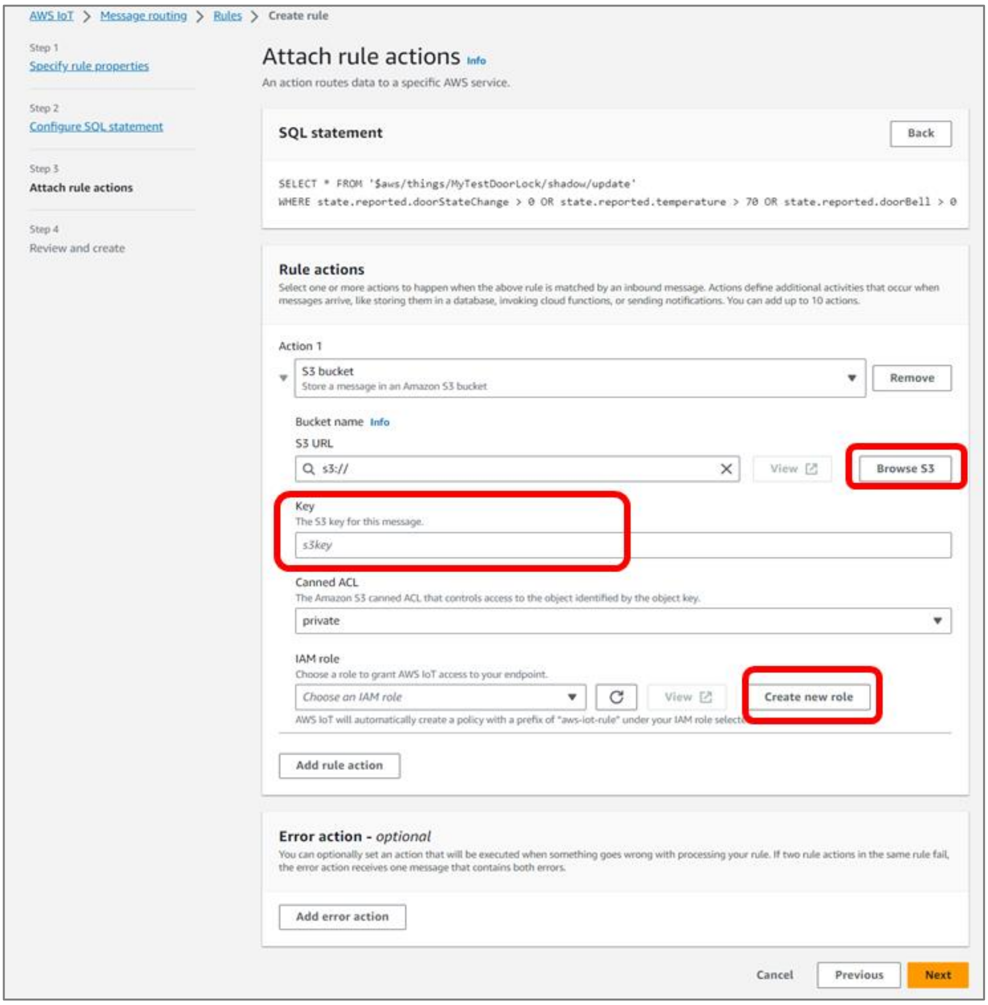
```
SELECT * FROM '$aws/things/MyTestDoorLock/shadow/update'
WHERE state.reported.doorStateChange > 0 OR state.reported.
    temperature > 70 OR state.reported.doorBell > 0
```

Rule actions
Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. You can add up to 10 actions.

Action 1
Choose an action

Figure 25. Attach rule actions

5. Click **Browse S3** and in the Key field, enter **${timestamp()}**. Then, click **Create new role**.



AWS IoT > Message routing > Rules > Create rule

Step 1
Specify rule properties

Step 2
Configure SQL statement

Step 3
Attach rule actions

Step 4
Review and create

Attach rule actions Info

An action routes data to a specific AWS service.

SQL statement

```
SELECT * FROM '$aws/things/MyTestDoorLock/shadow/update'
WHERE state.reported.doorStateChange > 0 OR state.reported.temperature > 70 OR state.reported.doorBell > 0
```

Rule actions

Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. You can add up to 10 actions.

Action 1

S3 bucket
Store a message in an Amazon S3 bucket

Bucket name Info

S3 URL: s3://

Key
The S3 key for this message.
s3key

Canned ACL
The Amazon S3 canned ACL that controls access to the object identified by the object key.
private

IAM role
Choose a role to grant AWS IoT access to your endpoint.
Choose an IAM role

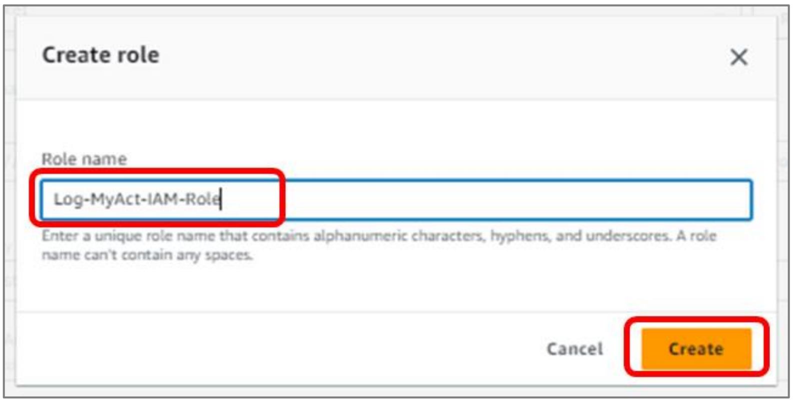
AWS IoT will automatically create a policy with a prefix of "aws-iot-rule" under your IAM role selecte

Error action - *optional*

You can optionally set an action that will be executed when something goes wrong with processing your rule. If two rule actions in the same rule fail, the error action receives one message that contains both errors.

Figure 26. Attach rule actions (continued)

6. In the **Role name** field, enter an IAM role name and click **Create**.



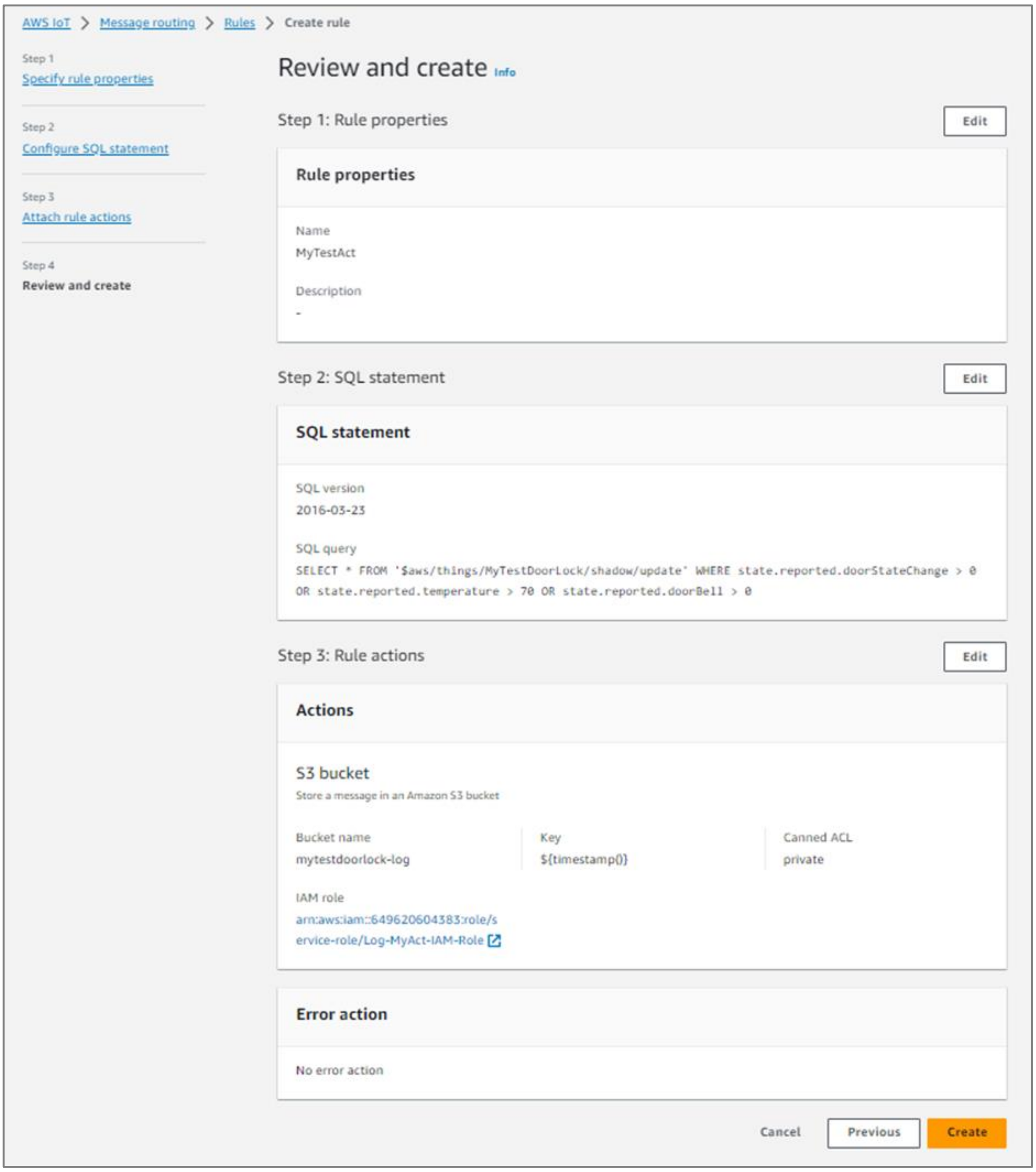
Create role

Role name
Log-MyAct-IAM-Role

Enter a unique role name that contains alphanumeric characters, hyphens, and underscores. A role name can't contain any spaces.

Figure 27. Create IAM role to save log files

7. Review the entered information and click **Create**.



Review and create Info

Step 1: Rule properties

Rule properties

Name
MyTestAct

Description
-

Step 2: SQL statement

SQL statement

SQL version
2016-03-23

SQL query

```
SELECT * FROM '$aws/things/MyTestDoorLock/shadow/update' WHERE state.reported.doorStateChange > 0
OR state.reported.temperature > 70 OR state.reported.doorBell > 0
```

Step 3: Rule actions

Actions

S3 bucket
Store a message in an Amazon S3 bucket

Bucket name	Key	Canned ACL
mytestdoorlock-log	${timestamp()}	private

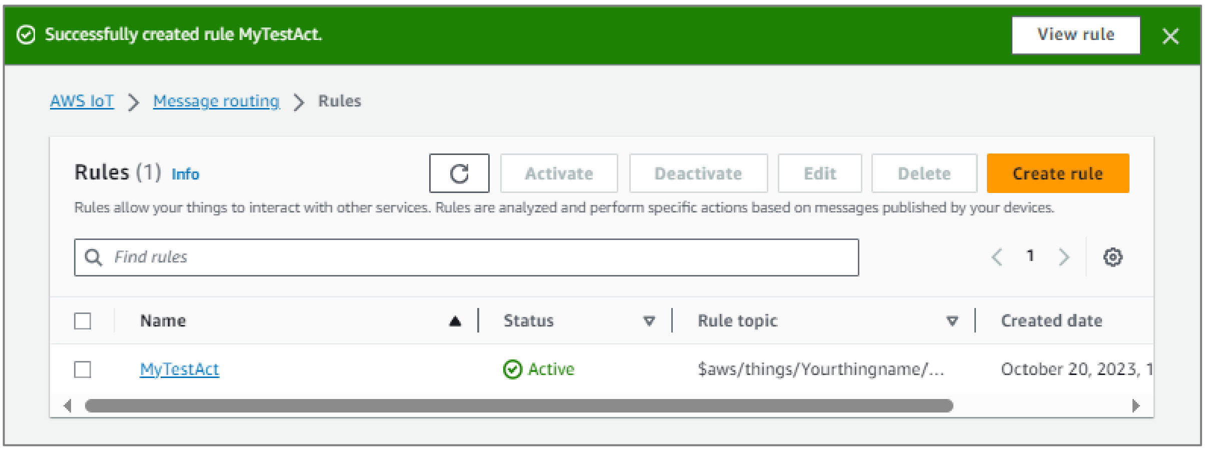
IAM role
arn:aws:iam::649620604383:role/service-role/Log-MyAct-IAM-Role

Error action

No error action

Figure 28. Review rules

8. The created rules should appear in the list of policies.



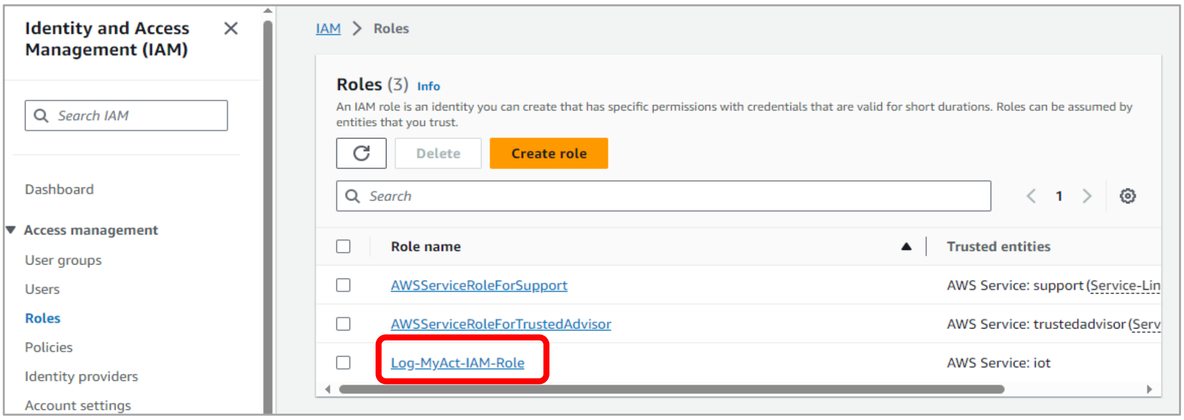
Successfully created rule MyTestAct.

Rules allow your things to interact with other services. Rules are analyzed and perform specific actions based on messages published by your devices.

Name	Status	Rule topic	Created date
MyTestAct	Active	$aws/things/Yourthingname/...	October 20, 2023, 1

Figure 29. Created rule

9. Go to the IAM console and select **Roles** and review the created roles.

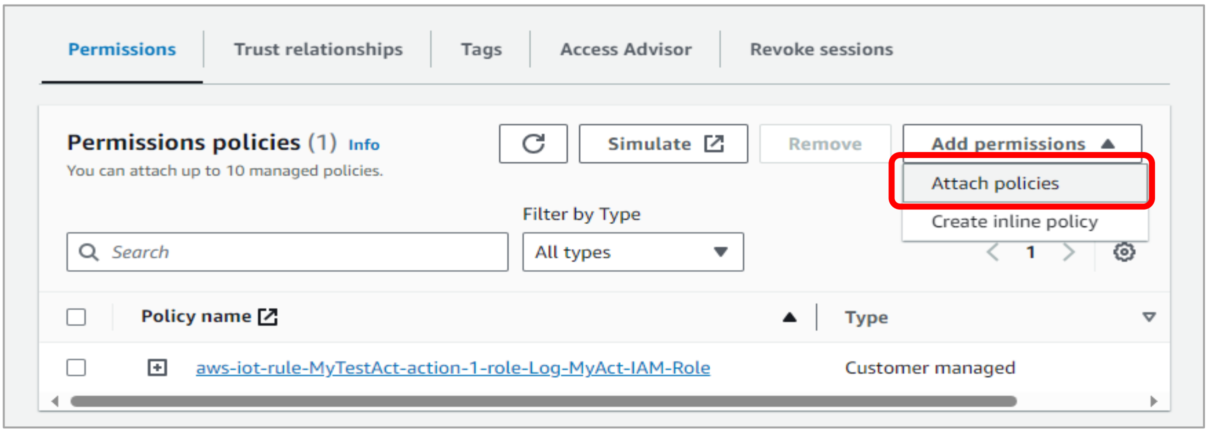


An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities
AWSServiceRoleForSupport	AWS Service: support (Service-Lin
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Serv
Log-MyAct-IAM-Role	AWS Service: iot

Figure 30. Created role

10. Choose the created role name and click **Attach policies**.

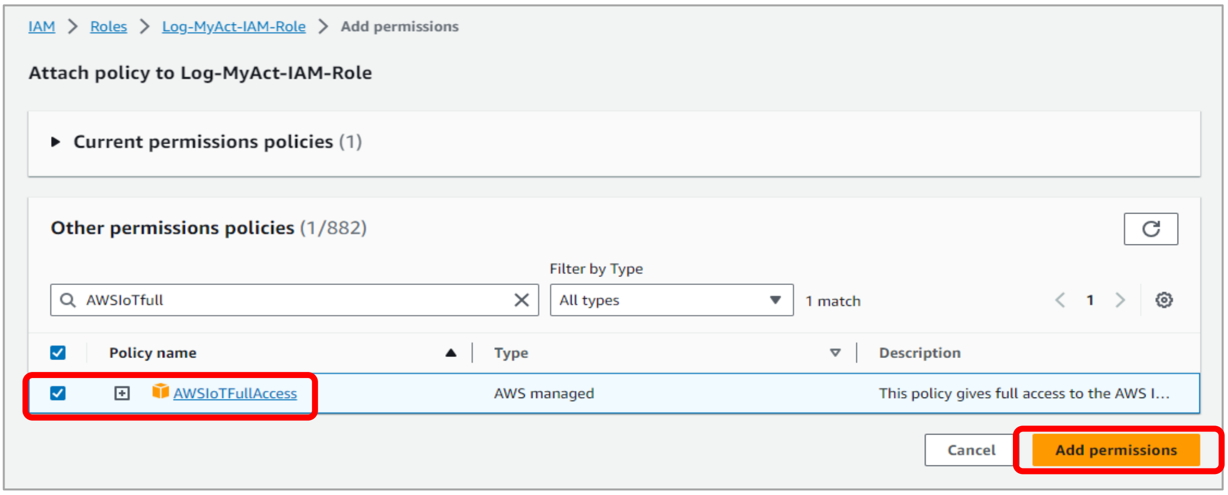


You can attach up to 10 managed policies.

Policy name	Type
aws-iot-rule-MyTestAct-action-1-role-Log-MyAct-IAM-Role	Customer managed

Figure 31. Attach policy to role

11. Search for the policy name of **AWSIoTFullAccess** and click **Add permissions**. Do the same thing for **AmazonS3FullAccess.**



IAM > Roles > Log-MyAct-IAM-Role > Add permissions

Attach policy to Log-MyAct-IAM-Role

▸ Current permissions policies (1)

Other permissions policies (1/882)

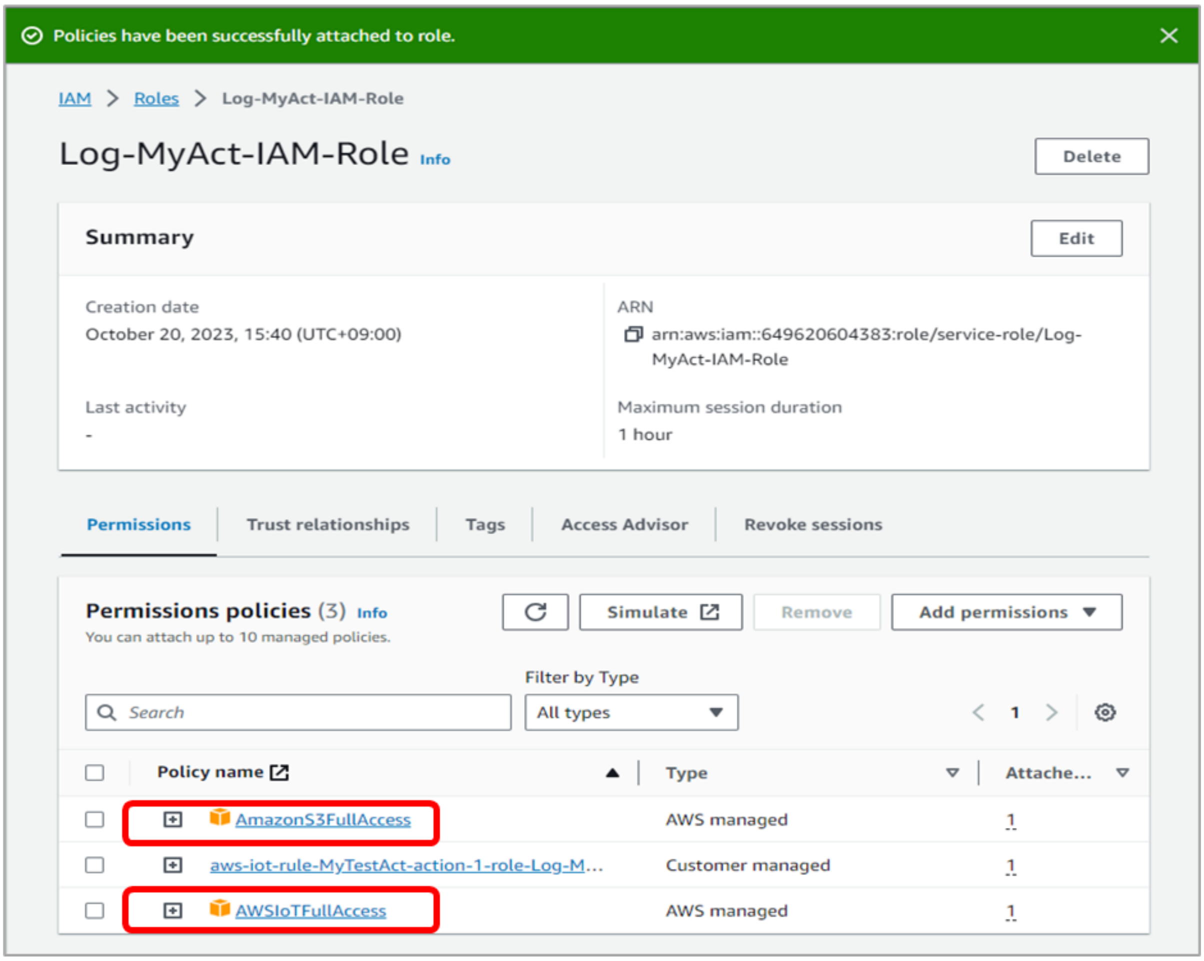
Search: AWSIoTfull | Filter by Type: All types | 1 match

☑	Policy name	Type	Description
☑	AWSIoTFullAccess	AWS managed	This policy gives full access to the AWS I…

Figure 32. AWSIoTFullAccess policy

NOTE
AWSIoTFullAccess and AmazonS3FullAccess policies are not recommended for production.

When the policies are added, the execution roles should look like Figure 33.



Policies have been successfully attached to role.

IAM > Roles > Log-MyAct-IAM-Role

Log-MyAct-IAM-Role Info

Summary

Creation date: October 20, 2023, 15:40 (UTC+09:00)

ARN: arn:aws:iam::649620604383:role/service-role/Log-MyAct-IAM-Role

Last activity: -

Maximum session duration: 1 hour

Permissions policies (3) Info

You can attach up to 10 managed policies.

Filter by Type: All types

□	Policy name	Type	Attache…
□	AmazonS3FullAccess	AWS managed	1
□	aws-iot-rule-MyTestAct-action-1-role-Log-M…	Customer managed	1
□	AWSIoTFullAccess	AWS managed	1

Figure 33. Attached policies

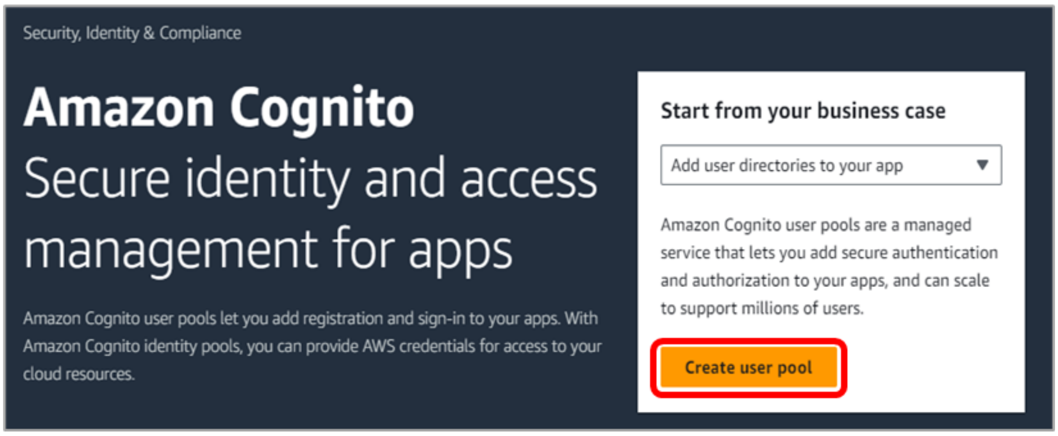
6.1.3 Configure Amazon Cognito

Amazon Cognito provides authentication, authorization, and user management for web and mobile apps. You can sign in directly with a username and password or through a third party such as Facebook, Amazon, or Google.

The two main components of Amazon Cognito are **user pools** and **identity pools**. User pools are directory of users that provide sign-up and sign-in options for app users. Identity pools provide AWS credentials to grant users access to other AWS services. Identity pools and user pools can be used separately or together. For more information, visit AWS Docs site (https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html).

6.1.3.1 Create User Pools

1. Go to the Amazon Cognito console and click **Create user pool**.



Security, Identity & Compliance

Amazon Cognito

Secure identity and access management for apps

Amazon Cognito user pools let you add registration and sign-in to your apps. With Amazon Cognito identity pools, you can provide AWS credentials for access to your cloud resources.

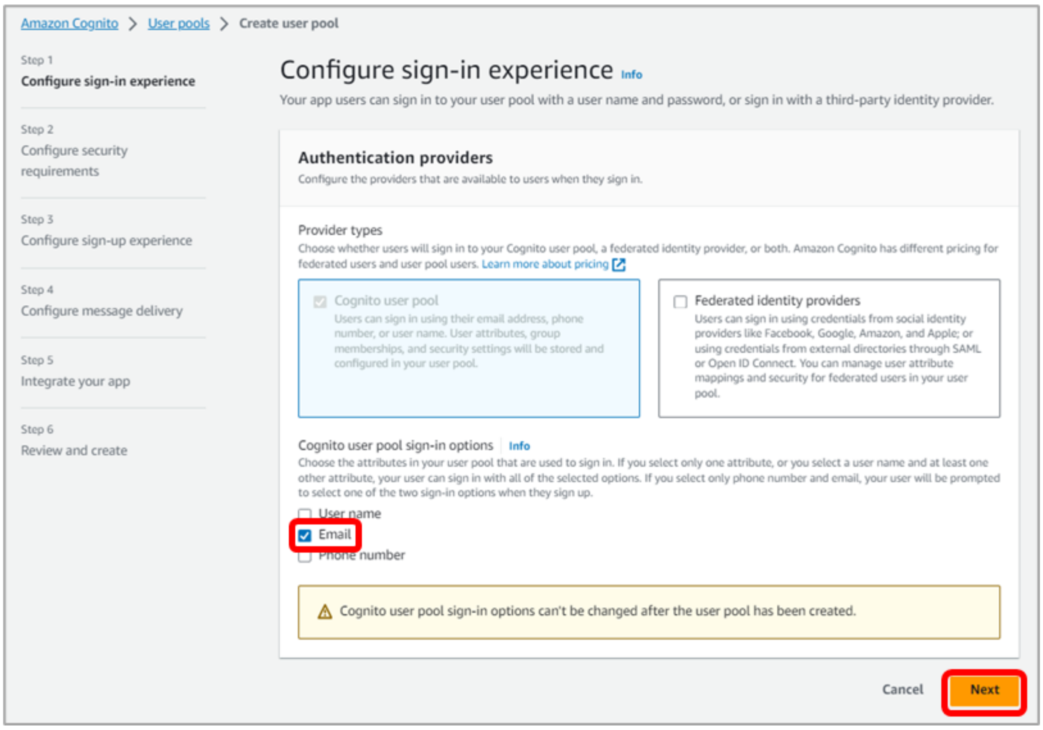
Start from your business case

Add user directories to your app ▼

Amazon Cognito user pools are a managed service that lets you add secure authentication and authorization to your apps, and can scale to support millions of users.

Figure 34. Create user pool

2. On the **Configure sign-in experience** page, select the **Email** checkbox, and click **Next**.



Amazon Cognito > User pools > Create user pool

Step 1
Configure sign-in experience

Step 2
Configure security requirements

Step 3
Configure sign-up experience

Step 4
Configure message delivery

Step 5
Integrate your app

Step 6
Review and create

Configure sign-in experience Info

Your app users can sign in to your user pool with a user name and password, or sign in with a third-party identity provider.

Authentication providers

Configure the providers that are available to users when they sign in.

Provider types

Choose whether users will sign in to your Cognito user pool, a federated identity provider, or both. Amazon Cognito has different pricing for federated users and user pool users. Learn more about pricing

☑ Cognito user pool
Users can sign in using their email address, phone number, or user name. User attributes, group memberships, and security settings will be stored and configured in your user pool.

□ Federated identity providers
Users can sign in using credentials from social identity providers like Facebook, Google, Amazon, and Apple; or using credentials from external directories through SAML or Open ID Connect. You can manage user attribute mappings and security for federated users in your user pool.

Cognito user pool sign-in options Info

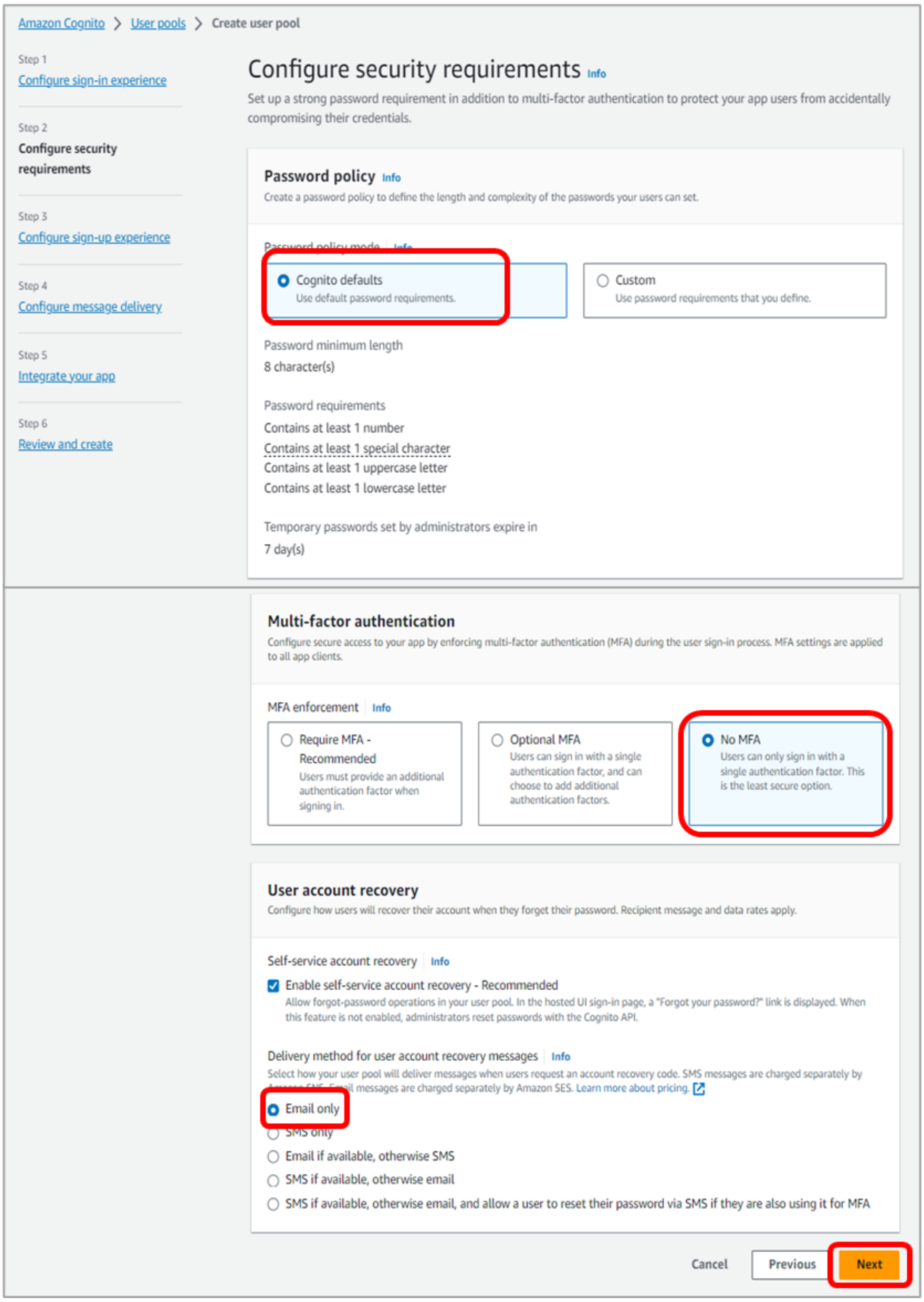
Choose the attributes in your user pool that are used to sign in. If you select only one attribute, or you select a user name and at least one other attribute, your user can sign in with all of the selected options. If you select only phone number and email, your user will be prompted to select one of the two sign-in options when they sign up.

□ User name
☑ Email
□ Phone number

⚠ Cognito user pool sign-in options can't be changed after the user pool has been created.

Figure 35. Configure sign-in options

3. Select **Cognito defaults** as password policy mode. Then, select **No MFA** and **Email only**, and click **Next**.



Amazon Cognito > User pools > Create user pool

Step 1
Configure sign-in experience

Step 2
Configure security requirements

Step 3
Configure sign-up experience

Step 4
Configure message delivery

Step 5
Integrate your app

Step 6
Review and create

Configure security requirements Info

Set up a strong password requirement in addition to multi-factor authentication to protect your app users from accidentally compromising their credentials.

Password policy Info

Create a password policy to define the length and complexity of the passwords your users can set.

Password policy mode | Info

● Cognito defaults
Use default password requirements.

○ Custom
Use password requirements that you define.

Password minimum length
8 character(s)

Password requirements
Contains at least 1 number
Contains at least 1 special character
Contains at least 1 uppercase letter
Contains at least 1 lowercase letter

Temporary passwords set by administrators expire in
7 day(s)

Multi-factor authentication

Configure secure access to your app by enforcing multi-factor authentication (MFA) during the user sign-in process. MFA settings are applied to all app clients.

MFA enforcement | Info

○ Require MFA - Recommended
Users must provide an additional authentication factor when signing in.

○ Optional MFA
Users can sign in with a single authentication factor, and can choose to add additional authentication factors.

● No MFA
Users can only sign in with a single authentication factor. This is the least secure option.

User account recovery

Configure how users will recover their account when they forget their password. Recipient message and data rates apply.

Self-service account recovery | Info

☑ Enable self-service account recovery - Recommended
Allow forgot-password operations in your user pool. In the hosted UI sign-in page, a "Forgot your password?" link is displayed. When this feature is not enabled, administrators reset passwords with the Cognito API.

Delivery method for user account recovery messages | Info

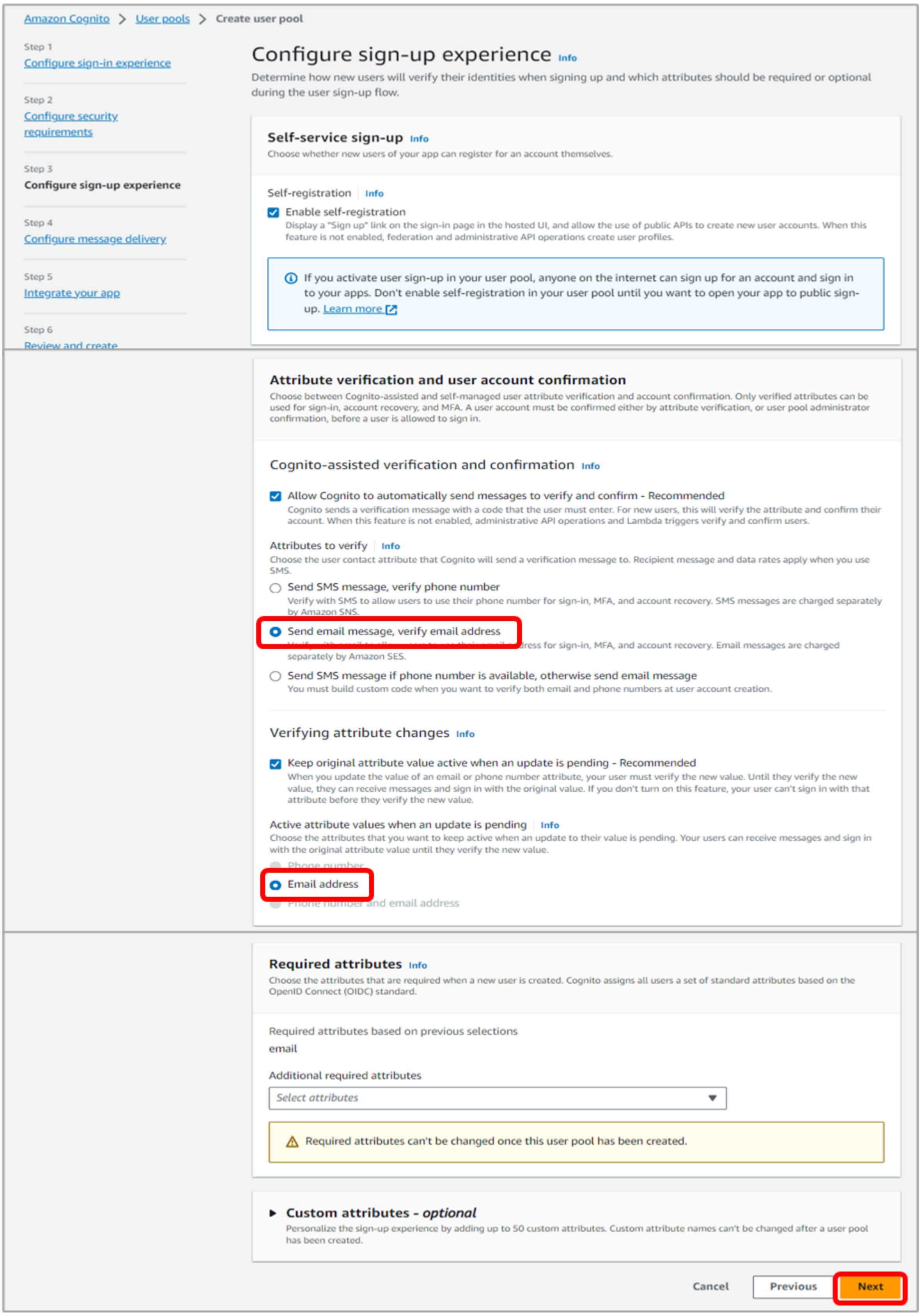
Select how your user pool will deliver messages when users request an account recovery code. SMS messages are charged separately by Amazon SNS. Email messages are charged separately by Amazon SES. Learn more about pricing.

● Email only
○ SMS only
○ Email if available, otherwise SMS
○ SMS if available, otherwise email
○ SMS if available, otherwise email, and allow a user to reset their password via SMS if they are also using it for MFA

Cancel | Previous | Next

Figure 36. Configure security requirements

4. Configure sign-up as shown in Figure 37.



Amazon Cognito > User pools > Create user pool

Step 1
Configure sign-in experience

Step 2
Configure security requirements

Step 3
Configure sign-up experience

Step 4
Configure message delivery

Step 5
Integrate your app

Step 6
Review and create

Configure sign-up experience Info

Determine how new users will verify their identities when signing up and which attributes should be required or optional during the user sign-up flow.

Self-service sign-up Info

Choose whether new users of your app can register for an account themselves.

Self-registration | Info

☑ Enable self-registration
Display a "Sign up" link on the sign-in page in the hosted UI, and allow the use of public APIs to create new user accounts. When this feature is not enabled, federation and administrative API operations create user profiles.

ⓘ If you activate user sign-up in your user pool, anyone on the internet can sign up for an account and sign in to your apps. Don't enable self-registration in your user pool until you want to open your app to public sign-up. Learn more

Attribute verification and user account confirmation

Choose between Cognito-assisted and self-managed user attribute verification and account confirmation. Only verified attributes can be used for sign-in, account recovery, and MFA. A user account must be confirmed either by attribute verification, or user pool administrator confirmation, before a user is allowed to sign in.

Cognito-assisted verification and confirmation Info

☑ Allow Cognito to automatically send messages to verify and confirm - Recommended
Cognito sends a verification message with a code that the user must enter. For new users, this will verify the attribute and confirm their account. When this feature is not enabled, administrative API operations and Lambda triggers verify and confirm users.

Attributes to verify | Info
Choose the user contact attribute that Cognito will send a verification message to. Recipient message and data rates apply when you use SMS.

○ Send SMS message, verify phone number
Verify with SMS to allow users to use their phone number for sign-in, MFA, and account recovery. SMS messages are charged separately by Amazon SNS.

● Send email message, verify email address
Verify with email to allow users to use their email address for sign-in, MFA, and account recovery. Email messages are charged separately by Amazon SES.

○ Send SMS message if phone number is available, otherwise send email message
You must build custom code when you want to verify both email and phone numbers at user account creation.

Verifying attribute changes Info

☑ Keep original attribute value active when an update is pending - Recommended
When you update the value of an email or phone number attribute, your user must verify the new value. Until they verify the new value, they can receive messages and sign in with the original value. If you don't turn on this feature, your user can't sign in with that attribute before they verify the new value.

Active attribute values when an update is pending | Info
Choose the attributes that you want to keep active when an update to their value is pending. Your users can receive messages and sign in with the original attribute value until they verify the new value.

○ Phone number
● Email address
○ Phone number and email address

Required attributes Info

Choose the attributes that are required when a new user is created. Cognito assigns all users a set of standard attributes based on the OpenID Connect (OIDC) standard.

Required attributes based on previous selections
email

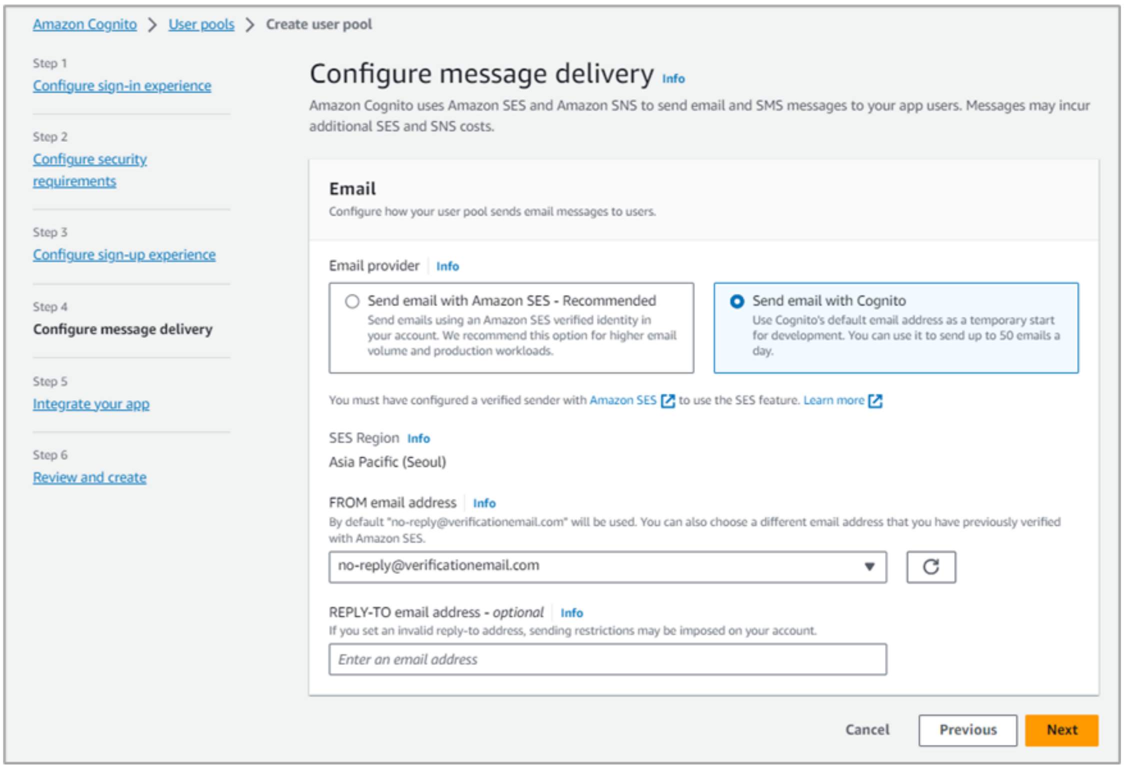
Additional required attributes
Select attributes ▼

⚠ Required attributes can't be changed once this user pool has been created.

▸ Custom attributes - *optional*
Personalize the sign-up experience by adding up to 50 custom attributes. Custom attribute names can't be changed after a user pool has been created.

Figure 37. Configure sign-up experience

5. Select **Send email with Cognito**.



Amazon Cognito > User pools > Create user pool

Step 1 Configure sign-in experience

Step 2 Configure security requirements

Step 3 Configure sign-up experience

Step 4 **Configure message delivery**

Step 5 Integrate your app

Step 6 Review and create

Configure message delivery Info

Amazon Cognito uses Amazon SES and Amazon SNS to send email and SMS messages to your app users. Messages may incur additional SES and SNS costs.

Email

Configure how your user pool sends email messages to users.

Email provider Info

○ Send email with Amazon SES - Recommended
Send emails using an Amazon SES verified identity in your account. We recommend this option for higher email volume and production workloads.

● Send email with Cognito
Use Cognito's default email address as a temporary start for development. You can use it to send up to 50 emails a day.

You must have configured a verified sender with Amazon SES to use the SES feature. Learn more

SES Region Info
Asia Pacific (Seoul)

FROM email address Info
By default "no-reply@verificationemail.com" will be used. You can also choose a different email address that you have previously verified with Amazon SES.

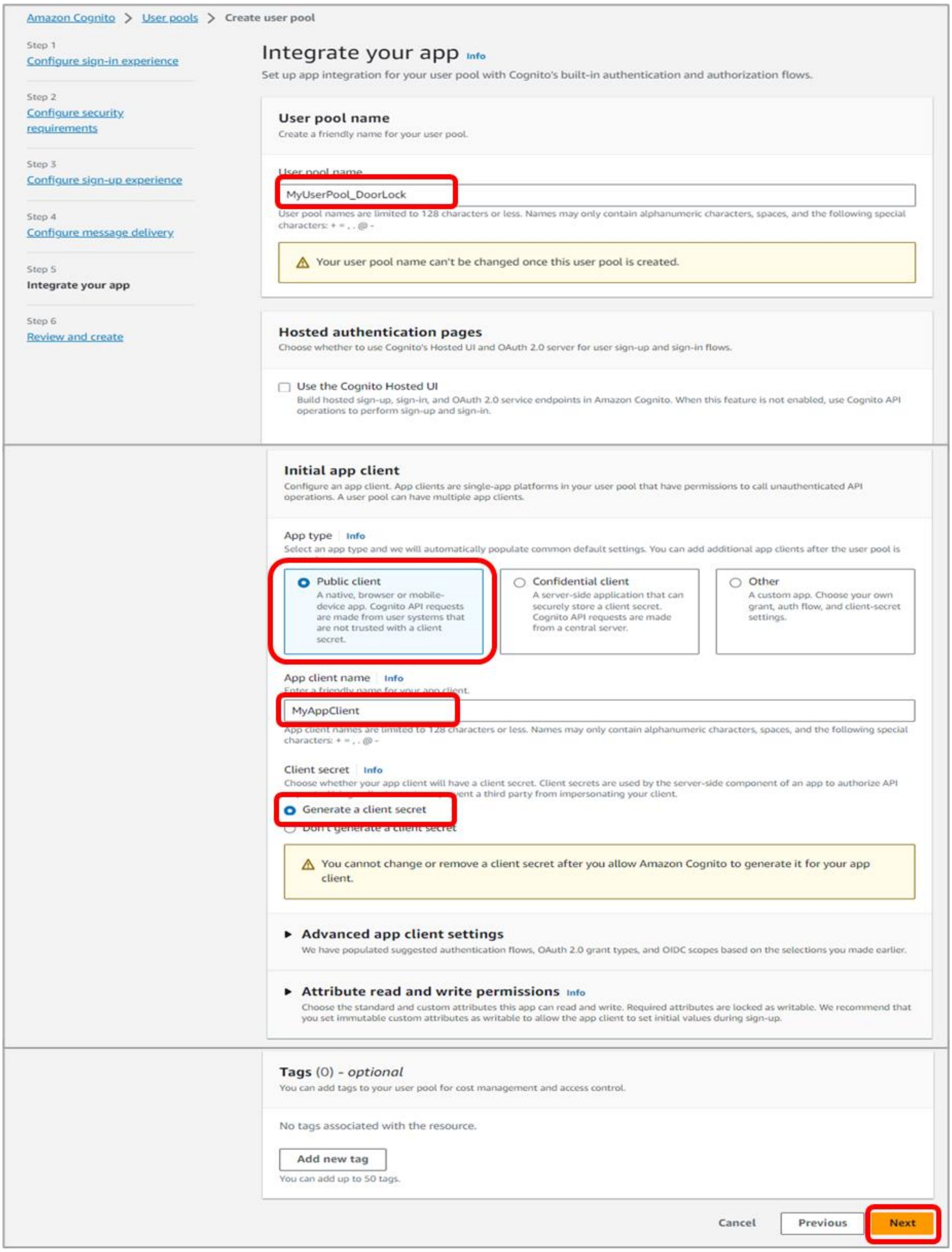
no-reply@verificationemail.com

REPLY-TO email address - *optional* Info
If you set an invalid reply-to address, sending restrictions may be imposed on your account.

Enter an email address

Figure 38. Configure message delivery

6. On the **Integrate your app** page, enter required items as shown in Figure 39 and click **Next**.



Amazon Cognito > User pools > Create user pool

Step 1
Configure sign-in experience

Step 2
Configure security requirements

Step 3
Configure sign-up experience

Step 4
Configure message delivery

Step 5
Integrate your app

Step 6
Review and create

Integrate your app Info

Set up app integration for your user pool with Cognito's built-in authentication and authorization flows.

User pool name

Create a friendly name for your user pool.

User pool name

MyUserPool_DoorLock

User pool names are limited to 128 characters or less. Names may only contain alphanumeric characters, spaces, and the following special characters: + = , . @ -

⚠ Your user pool name can't be changed once this user pool is created.

Hosted authentication pages

Choose whether to use Cognito's Hosted UI and OAuth 2.0 server for user sign-up and sign-in flows.

□ Use the Cognito Hosted UI
Build hosted sign-up, sign-in, and OAuth 2.0 service endpoints in Amazon Cognito. When this feature is not enabled, use Cognito API operations to perform sign-up and sign-in.

Initial app client

Configure an app client. App clients are single-app platforms in your user pool that have permissions to call unauthenticated API operations. A user pool can have multiple app clients.

App type Info

Select an app type and we will automatically populate common default settings. You can add additional app clients after the user pool is

● Public client
A native, browser or mobile-device app. Cognito API requests are made from user systems that are not trusted with a client secret.

○ Confidential client
A server-side application that can securely store a client secret. Cognito API requests are made from a central server.

○ Other
A custom app. Choose your own grant, auth flow, and client-secret settings.

App client name Info

Enter a friendly name for your app client.

MyAppClient

App client names are limited to 128 characters or less. Names may only contain alphanumeric characters, spaces, and the following special characters: + = , . @ -

Client secret Info

Choose whether your app client will have a client secret. Client secrets are used by the server-side component of an app to authorize API ent a third party from impersonating your client.

● Generate a client secret

○ Don't generate a client secret

⚠ You cannot change or remove a client secret after you allow Amazon Cognito to generate it for your app client.

▶ Advanced app client settings
We have populated suggested authentication flows, OAuth 2.0 grant types, and OIDC scopes based on the selections you made earlier.

▶ Attribute read and write permissions Info
Choose the standard and custom attributes this app can read and write. Required attributes are locked as writable. We recommend that you set immutable custom attributes as writable to allow the app client to set initial values during sign-up.

Tags (0) - *optional*

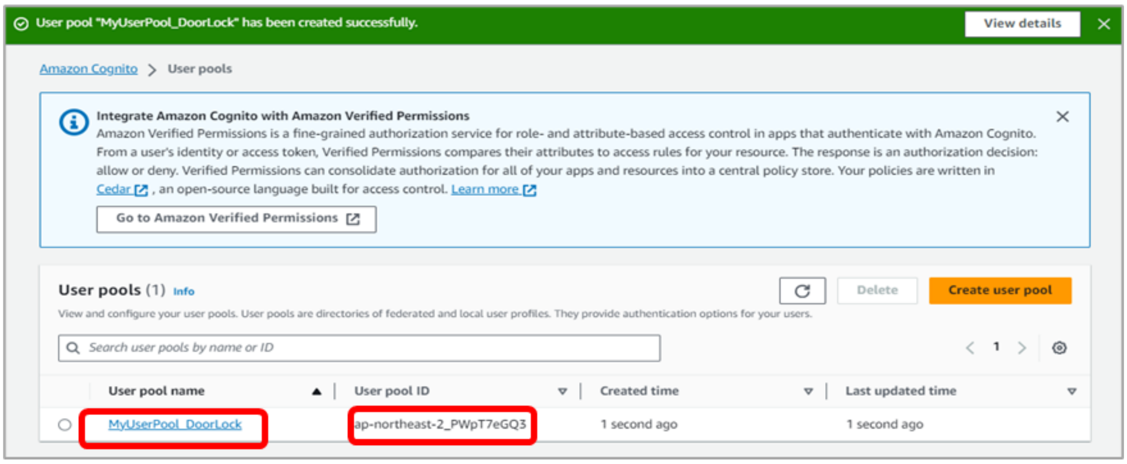
You can add tags to your user pool for cost management and access control.

No tags associated with the resource.

You can add up to 50 tags.

Figure 39. Integrate app client

7. On the **Review and create** page, review the entered information, and click **Create user pool**. Then, the created user pool should appear in the list.



User pool "MyUserPool_DoorLock" has been created successfully.

Integrate Amazon Cognito with Amazon Verified Permissions

Amazon Verified Permissions is a fine-grained authorization service for role- and attribute-based access control in apps that authenticate with Amazon Cognito. From a user's identity or access token, Verified Permissions compares their attributes to access rules for your resource. The response is an authorization decision: allow or deny. Verified Permissions can consolidate authorization for all of your apps and resources into a central policy store. Your policies are written in Cedar, an open-source language built for access control. Learn more

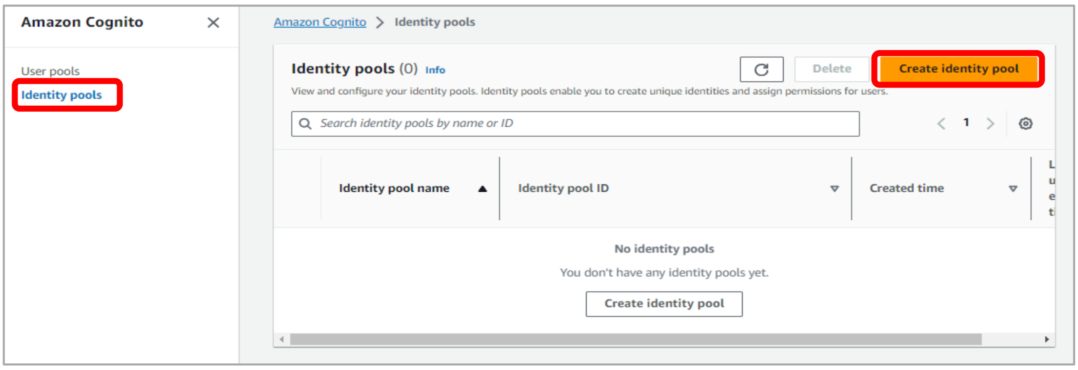
View and configure your user pools. User pools are directories of federated and local user profiles. They provide authentication options for your users.

	User pool name	User pool ID	Created time	Last updated time
○	MyUserPool_DoorLock	ap-northeast-2_PWpT7eGQ3	1 second ago	1 second ago

Figure 40. Created user pool

6.1.3.2 Create Identity Pools

1. Go to the Amazon Cognito console. Choose **Identity pools** and click **Create identity pool**.



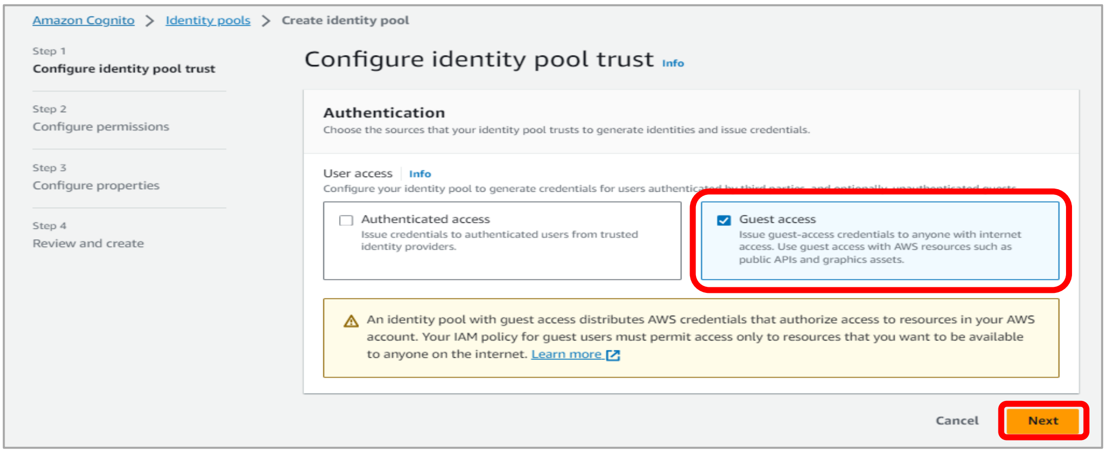
View and configure your identity pools. Identity pools enable you to create unique identities and assign permissions for users.

No identity pools

You don't have any identity pools yet.

Figure 41. Create identity pool

2. On the **Configure identity pool trust** page, select **Guest access** and click **Next**.



Amazon Cognito > Identity pools > Create identity pool
Step 1
Configure identity pool trust
Step 2
Configure permissions
Step 3
Configure properties
Step 4
Review and create

Configure identity pool trust Info

Authentication

Choose the sources that your identity pool trusts to generate identities and issue credentials.

User access Info

Configure your identity pool to generate credentials for users authenticat

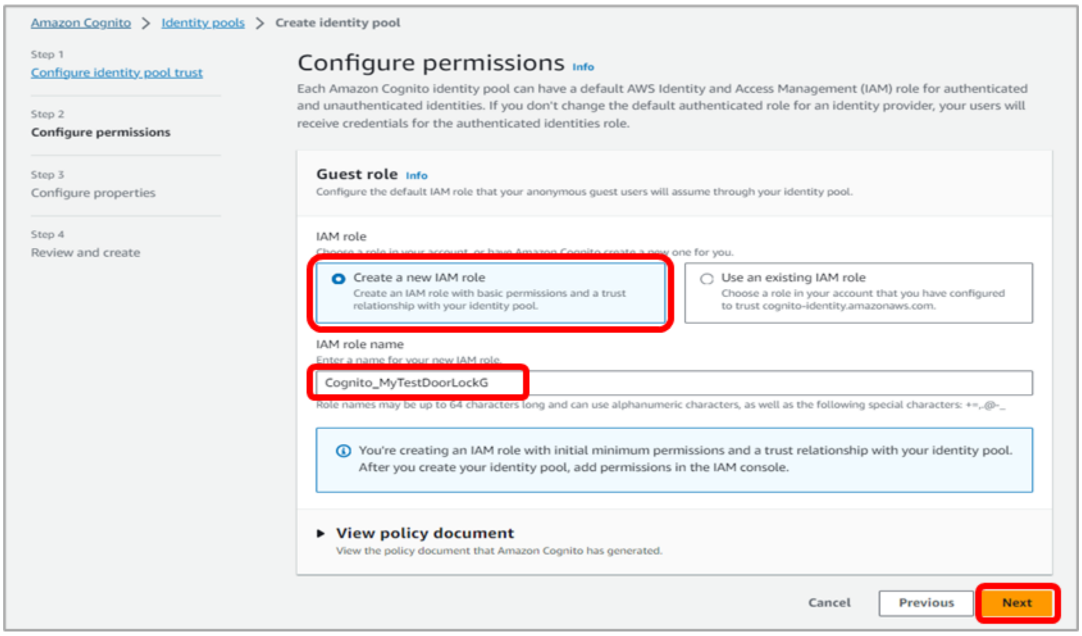
☐ Authenticated access
Issue credentials to authenticated users from trusted identity providers.

☑ Guest access
Issue guest-access credentials to anyone with internet access. Use guest access with AWS resources such as public APIs and graphics assets.

⚠ An identity pool with guest access distributes AWS credentials that authorize access to resources in your AWS account. Your IAM policy for guest users must permit access only to resources that you want to be available to anyone on the internet. Learn more

Figure 42. Create identity pool trust

3. On the **Configure permissions** page, select **Create a new IAM role,** enter an **IAM role name**, and then click **Next**.



Amazon Cognito > Identity pools > Create identity pool

Step 1
Configure identity pool trust

Step 2
Configure permissions

Step 3
Configure properties

Step 4
Review and create

Configure permissions Info

Each Amazon Cognito identity pool can have a default AWS Identity and Access Management (IAM) role for authenticated and unauthenticated identities. If you don't change the default authenticated role for an identity provider, your users will receive credentials for the authenticated identities role.

Guest role Info

Configure the default IAM role that your anonymous guest users will assume through your identity pool.

IAM role

Choose a role in your account, or have Amazon Cognito create a new one for you.

● Create a new IAM role
Create an IAM role with basic permissions and a trust relationship with your identity pool.

○ Use an existing IAM role
Choose a role in your account that you have configured to trust cognito-identity.amazonaws.com.

IAM role name

Enter a name for your new IAM role.

Cognito_MyTestDoorLockG

Role names may be up to 64 characters long and can use alphanumeric characters, as well as the following special characters: +=,.@-_

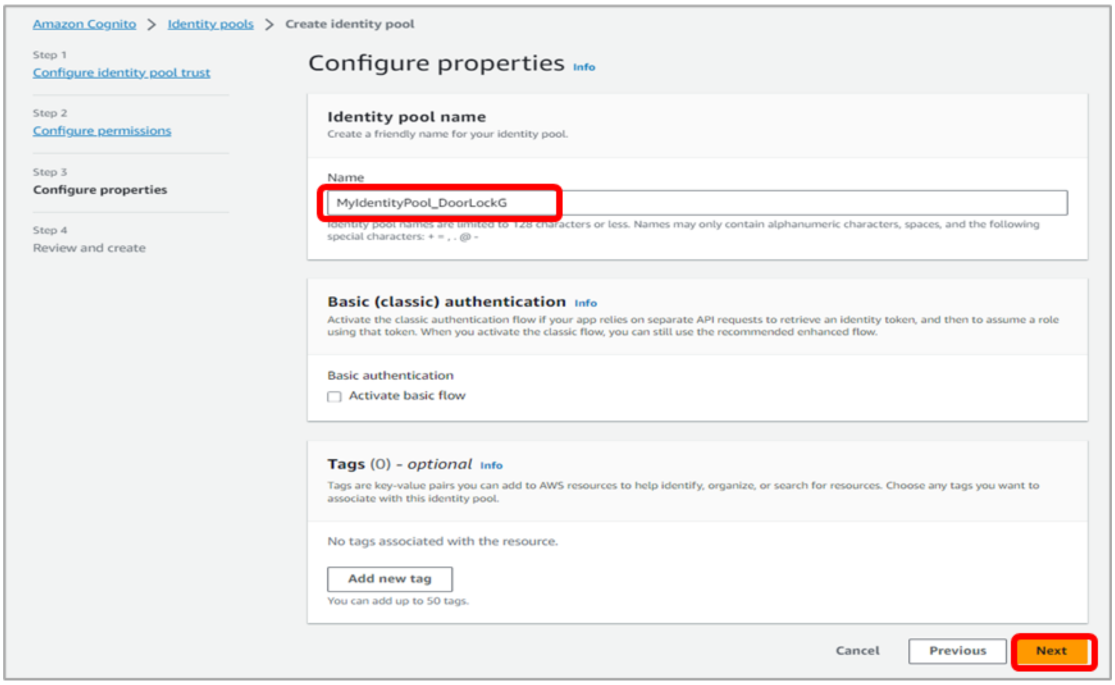
You're creating an IAM role with initial minimum permissions and a trust relationship with your identity pool. After you create your identity pool, add permissions in the IAM console.

▸ View policy document

View the policy document that Amazon Cognito has generated.

Figure 43. Configure permissions

4. Enter an Identity pool name and click **Next**.



Amazon Cognito > Identity pools > Create identity pool

Step 1
Configure identity pool trust

Step 2
Configure permissions

Step 3
Configure properties

Step 4
Review and create

Configure properties Info

Identity pool name

Create a friendly name for your identity pool.

Name

MyIdentityPool_DoorLockG

Identity pool names are limited to 128 characters or less. Names may only contain alphanumeric characters, spaces, and the following special characters: + = , . @ -

Basic (classic) authentication Info

Activate the classic authentication flow if your app relies on separate API requests to retrieve an identity token, and then to assume a role using that token. When you activate the classic flow, you can still use the recommended enhanced flow.

Basic authentication

□ Activate basic flow

Tags (0) - *optional* Info

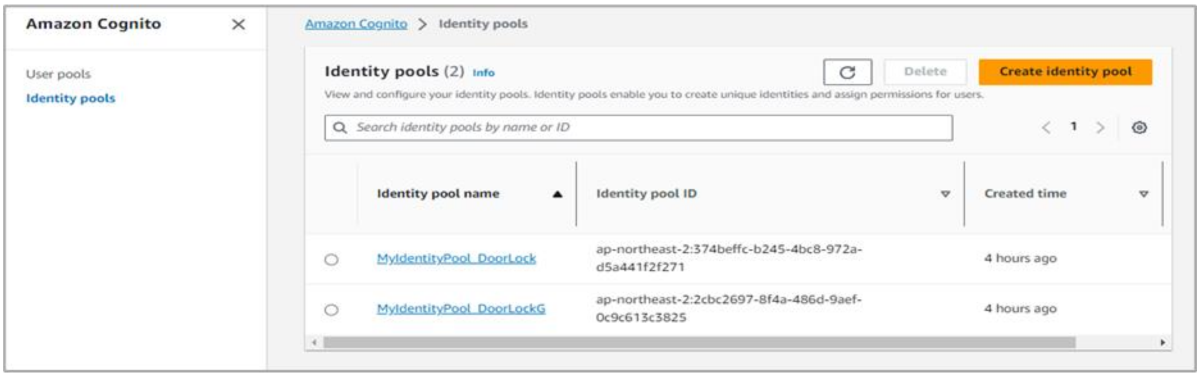
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this identity pool.

No tags associated with the resource.

You can add up to 50 tags.

Figure 44. Configure properties

5. Review the selected items and click **Create identity pool**. Then, the created identity pools appear in the list.



View and configure your identity pools. Identity pools enable you to create unique identities and assign permissions for users.

	Identity pool name	Identity pool ID	Created time
○	MyIdentityPool_DoorLock	ap-northeast-2:374beffc-b245-4bc8-972a-d5a441f2f271	4 hours ago
○	MyIdentityPool_DoorLockG	ap-northeast-2:2cbc2697-8f4a-486d-9aef-0c9c613c3825	4 hours ago

Figure 45. Created identity pools

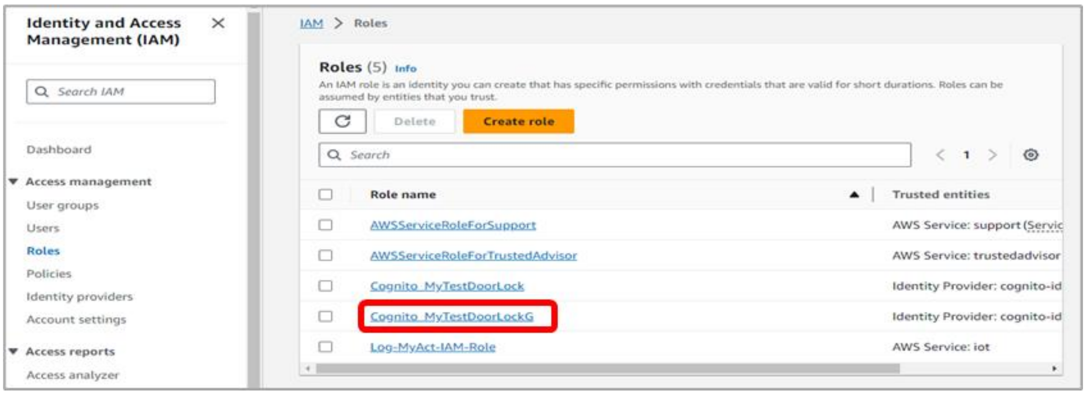
6.1.4 Set Up AWS IAM

While creating an identity pool, you should update the IAM roles that the users assume. When a user logs in to the app, Amazon Cognito generates temporary AWS credentials for the user. These temporary credentials are associated with a specific IAM role. The IAM role lets users define a set of permissions to access AWS resources. For more information, visit https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html.

- The roles in Cognito_MyTestDoorlockG are created automatically when the federation identity is created through Cognito Identity Pool.
- The device only needs an unauthorized role.

To set up AWS IAM:

1. Go to the IAM console and select **Cognito_MyTestDoorLockG**.



An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

□	Role name	Trusted entities
□	AWSServiceRoleForSupport	AWS Service: support (Servic
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor
□	Cognito_MyTestDoorLock	Identity Provider: cognito-id
□	Cognito_MyTestDoorLockG	Identity Provider: cognito-id
□	Log-MyAct-IAM-Role	AWS Service: iot

Figure 46. IAM role

2. Expand **Add permissions** and click **Attach policies**.

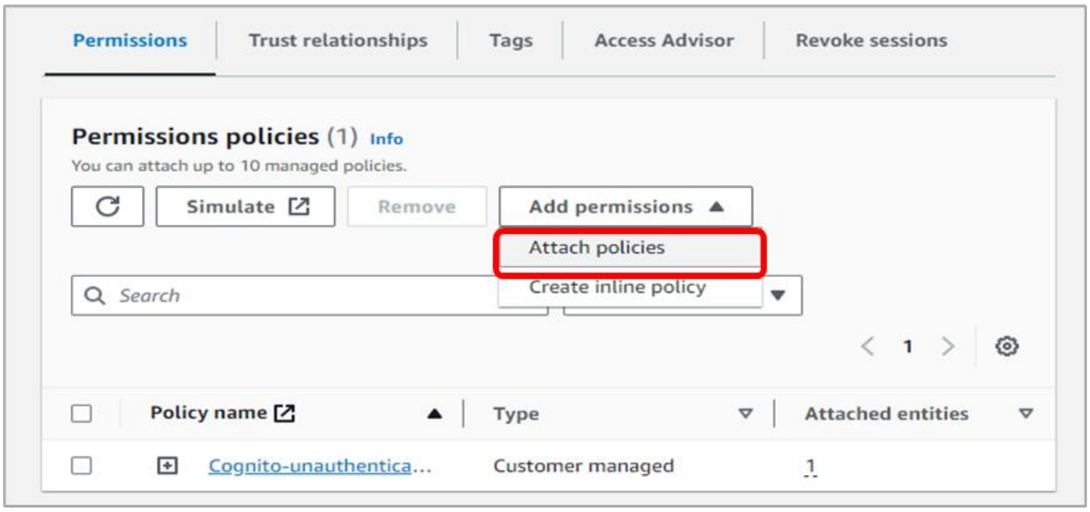


Figure 47. Attach policies

3. Search for the policy name of **AWSIoTFullAccess** and click **Add permissions**.

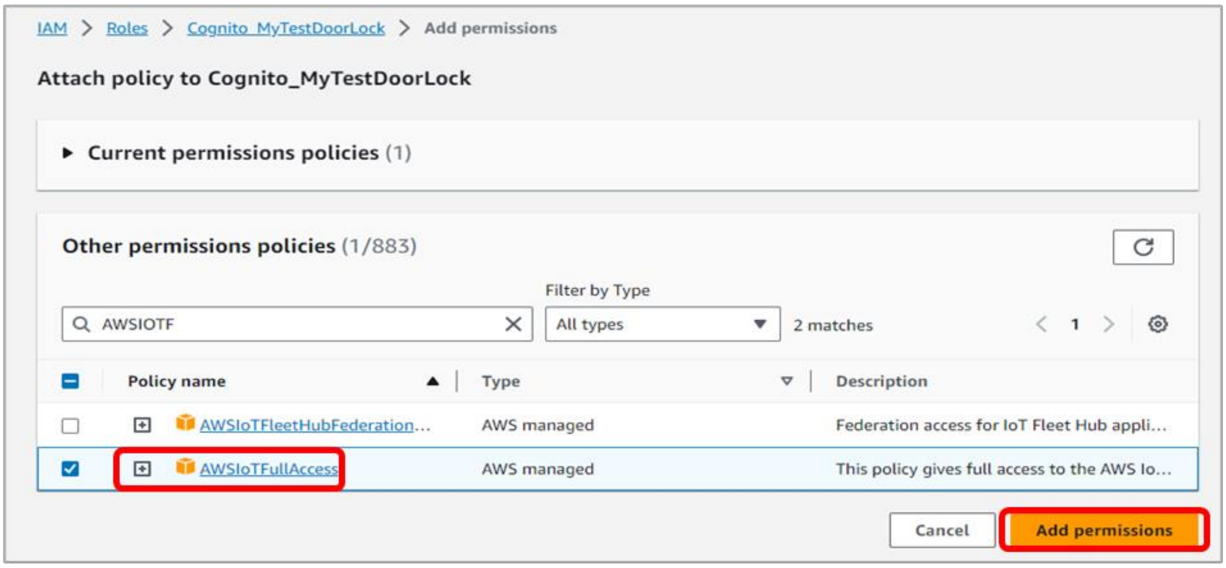

IAM > Roles > Cognito_MyTestDoorLock > Add permissions
Attach policy to Cognito_MyTestDoorLock
Current permissions policies (1)
Other permissions policies (1/883)
Filter by Type
AWSIOTF
All types
2 matches
1
Policy name
Type
Description
AWSIoTFleetHubFederation...
AWS managed
Federation access for IoT Fleet Hub appli...
AWSIoTFullAccess
AWS managed
This policy gives full access to the AWS Io...
Cancel
Add permissions

Figure 48. AWSIoTFullAccess policy

NOTE
AWSIoTFullAccess policy is not recommended for production.

4. Search for the policy name of **AmazonS3FullAccess** and click **Add permissions**.

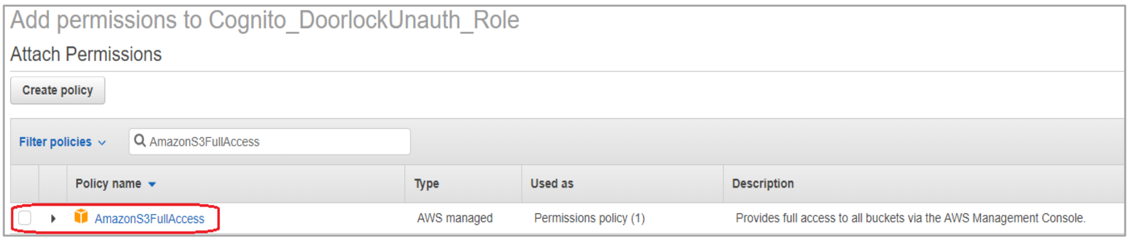
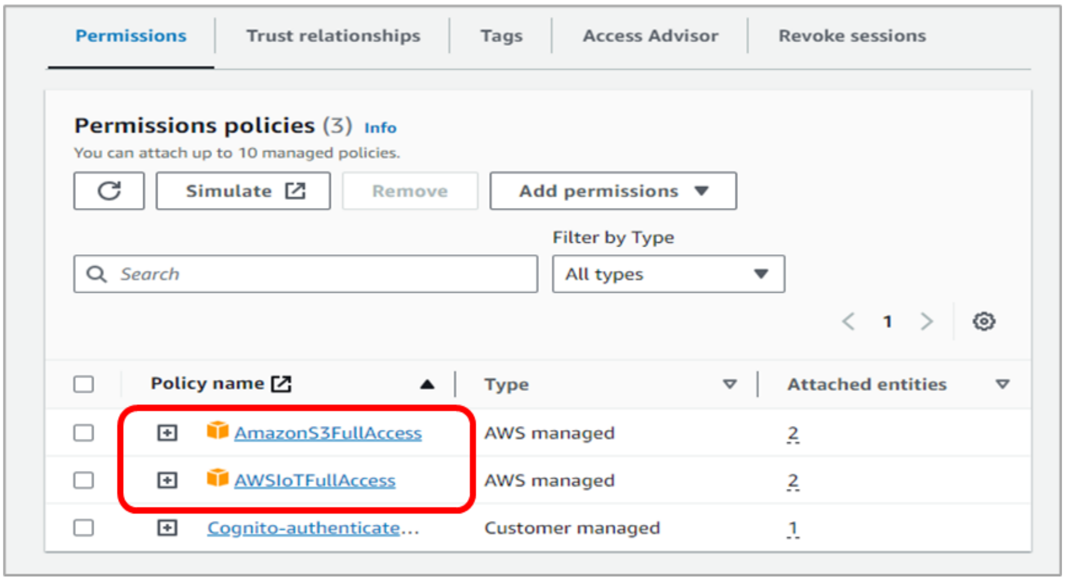


Figure 49. AmazonS3FullAccess policy

NOTE
AmazonS3FullAccess policy is not recommended for production.

5. The attached policies appear in the list.



☐	Policy name ▲	Type	Attached entities
☐	⊞ AmazonS3FullAccess	AWS managed	2
☐	⊞ AWSIoTFullAccess	AWS managed	2
☐	⊞ Cognito-authenticate…	Customer managed	1

Figure 50. Attached policies

6.1.5 Create Amazon S3 Bucket

Every object in Amazon S3 is stored in a bucket. Before storing data in Amazon S3, you need to create a bucket. To create S3 bucket:

1. In the Amazon S3 console, in the left navigation pane, select **Buckets**, and click **Create bucket**.
2. On the **Create bucket** page, in the **Bucket name** field, type a bucket name.
3. For **Region**, choose the AWS region where you want the bucket to reside, and click **Create bucket**.

   When Amazon S3 successfully creates the bucket, the console displays an empty bucket in the **Buckets** pane.

7. Build and Run Reference Application

Door lock reference application is available on the Renesas official website.

NOTE
Go to the Renesas website (https://www.renesas.com/us/en/products/wireless-connectivity/wi-fi/low-power-wi-fi), click Product Selector, check one of MOD devices, and scroll down to the Software Downloads section. Find "AWS IoT Reference" or type it in the search box, and then select the reference package and download.

For more detail, see DA16200 DA16600 FreeRTOS Getting Started Guide, Ref. [3].

7.1 Reference DA16200/DA16600 SDK Setting

7.1.1 Edit Endpoint

1. Change the AWS_USER_MQTT_HOST in the **app_aws_user_conf.h** file as follows:
   - AWS IoT > Settings

```
#define AWS_USER_MQTT_HOST    "(account-specific-prefix).iot.(aws-region).amazonaws.com"
```

2. Build the SDK and then update the image.

7.1.2 Edit Thing Name

7.1.2.1 Edit Thing Name with Console Command

You can directly rename things using the console commands without the need to build an SDK. Renesas recommends using console commands over other methods for changing a thing name.

If the test board is running, run the factory command first, and then proceed to NVRAM as follows.

```
[/DA16200] #
[/DA16200] # nvram
Command-List is changed, "NVRAM"
[/DA16200/NVRAM] #
[/DA16200/NVRAM] # setenv APP_THINGNAME USER_THING_NAME  // write user thing name
[/DA16200/NVRAM] # getenv                                // read user thing name
[/DA16200/NVRAM] # unsetenv APP_THINGNAME                // remove user thing name
[/DA16200/NVRAM] #
```

Then, complete the provisioning process.

```
[/DA16200] # nvram
    Command-List is changed, "NVRAM"
[/DA16200/NVRAM] # setenv APP_THINGNAME APP-DOORLOCK-1
[/DA16200/NVRAM] # getenv

Total length (411)
APP_THINGNAME (STR,15) ....... APP-DOORLOCK-1
N1_Profile (STR,02) .......... 1
N1_mode (STR,02) ............ 2
SYSMODE (STR,02) ............ 1
N1_ssid (STR,17) ............ "Renesas_DA16200"
N1_psk (STR,13) ............. "1234567890"
N1_proto (STR,04) ........... RSN
```

```
N1_key_mgmt (STR,08) ......... WPA-PSK
country_code (STR,03) ........ KR
1:IPADDR (STR,09) ............ 10.0.0.1
1:NETMASK (STR,14) ........... 255.255.255.0
1:GATEWAY (STR,09) ........... 10.0.0.1
1:DNSSVR (STR,08) ............ 8.8.8.8
USEDHCPD (STR,02) ............ 1
DHCPD_IPCNT (STR,03) ......... 10
DHCPD_TIME (STR,05) .......... 3600
DHCPD_S_IP (STR,09) .......... 10.0.0.2
DHCPD_E_IP (STR,10) .......... 10.0.0.11
DHCPD_DNS (STR,08) ........... 8.8.8.8
[/DA16200/NVRAM] #
```

7.1.2.2 Edit Thing Name in Configuration File

If the thing name does not exist in NVRAM, the predefined name located in the first header is stored in NVRAM. Change AWS_USER_MY_THING_NAME in the **app_thing_manager.h** file, then build the SDK and update the image:

```
/*
*   USER Thing name define
*   Generic SDK default : "DA16200"
*   AWS IOT default : "IOT-SENSOR-46" or "FAE-DOORLOCK-4" or "assigned_thing_name"
*/
#define APP_USER_MY_THING_NAME  "FAE-DOORLOCK-4"
```

7.1.3 Edit Image File Name for OTA

To test the OTA update, edit the **app_aws_user_conf.h** file in the DA16200 SDK and modify the file names to match the image file names that are uploaded to the Amazon S3 bucket.

```
#if defined(__BLE_COMBO_REF__)
#define RTOS_NAME                      "DA16600_FRTOS-GEN01.img"
#define BLE_NAME                               "DA16600_BLE_OTA.img"
#else
#define RTOS_NAME                      "DA16200_FRTOS-GEN01.img"
#endif
```

7.1.4 Connect Certificates to Thing

To authenticate the device with AWS IoT, the device must contain the **Root CA**, **Client Certificate,** and **Client Private Key**. For more information, see https://docs.aws.amazon.com/iot/latest/developerguide/iot-security-identity.html.

To add these certificates to the device, edit **app_aws_certi.h** and insert the certificates downloaded from AWS as follows:

```
#define democonfigROOT_CA_PEM   "-----BEGIN CERTIFICATE-----\n" \
"MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPmljZbyjANBgkqhkiG9w0BAQsF\n" \
"ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6\n" \
"b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTM4MDExNzAwMDAwMFowOTEL\n" \
. . .
"o/ufQJVtMVT8QtPHRh8jrdkPSHCa2XV4cdFyQzR1bldZwgJcJmApzyMZFo6IQ6XU\n" \
"5MsI+yMRQ+hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy\n" \
"rqXRfboQnoZsG4q5WTP468SQvvG5\n" \
"-----END CERTIFICATE-----\n"

#define democonfigCLIENT_CERTIFICATE_PEM        "-----BEGIN CERTIFICATE-----\n" \
"MIIDWjCCAkKgAwIBAgIVAIqSKvd/Qq2E9ZleQWN2Gk/iPw2GMA0GCSqGSIb3DQEB\n" \
"CwUAME0xSzBJBgNVBAsMQkFtYXpvbiBXZWIgU2VydmljZXMgTz1BbWF6b24uY29t\n" \
"IEluYy4gTD1TZWF0dGxlIFNUPVdhc2hpbmd0b24gQz1VUzAeFw0xODEyMDYwNjQw\n" \
. . .
"TcaCwbJQy2XprqPpBo3ZuWqmSi55uslXj+2B4XgPZutim++8J7DHQbfHAGZwiAFN\n" \
"90TNlhZBdI87Ga07p0db03KcBQs8dBMaABC0RK39LqJ5ZdQMT/Owx0+iO2Be7w30\n" \
"7o06zCQB2A0nmfvAR8gSuImIBfKz2I1xQX5+CO4wes8RH5pNIOK2QrKgr9NJkA==\n" \
"-----END CERTIFICATE-----\n"

#define democonfigCLIENT_PRIVATE_KEY_PEM        "-----BEGIN RSA PRIVATE KEY-----\n" \
"MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1I66YD0p62x3C8ITqSiC6B4iz\n" \
"ugk6n17/6cXF8odFAh6adTxet5tL5mGLgLnkYFtt7Iyj10T8hpxT1Yxp7TYZRblw\n" \
"Fl9fptPRi5KncVhs9sICqJEmvKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \
. . .
"EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy79O\n" \
"7ETTGfSAbksUljV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxoygGW0JvOyY\n" \
"kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA==\n" \
"-----END RSA PRIVATE KEY-----\n" \
```

7.2 Reference Application in DA16200/DA16600

The following components shown in Figure 51 are required to run the application in DA16200/DA16600 through an Internet connection and AWS IoT server:

- AWS IoT reference application package
- DA16200/DA1660 EVB
- Router: Connection to internet
- Mobile device: Android/iOS application
- AWS account.

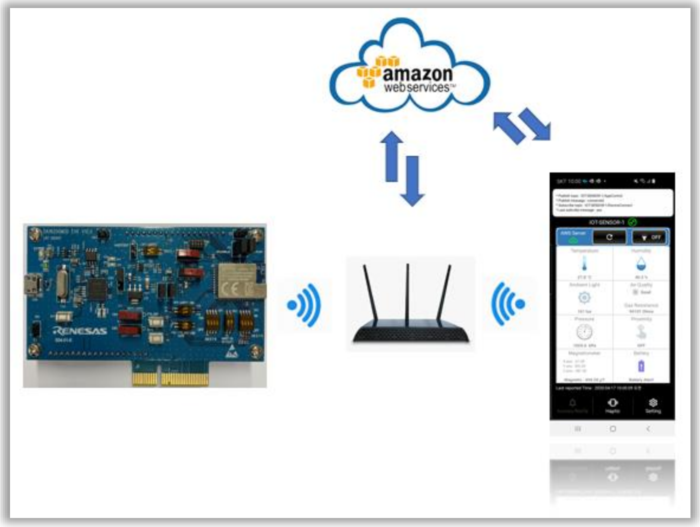


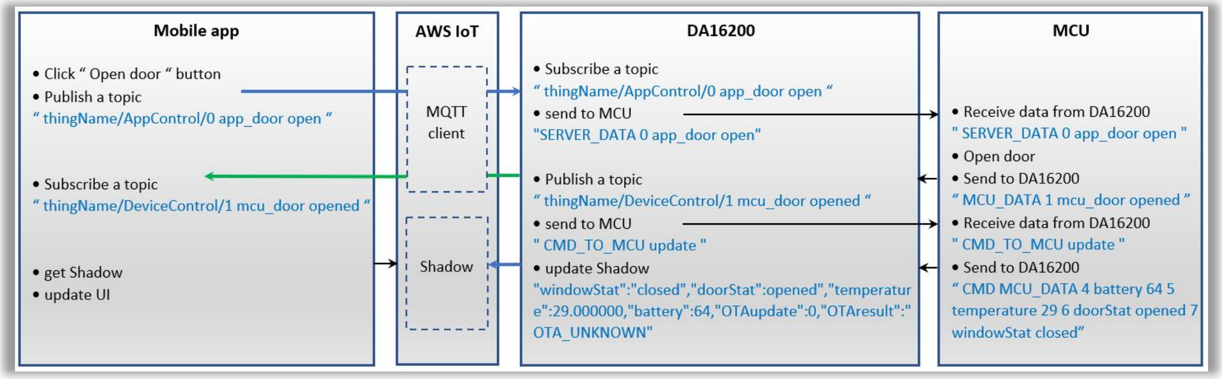
Figure 51. Architecture of AWS IoT

Install the mobile application by searching for **DA16200** or **DA16600** in the Google Pay Store or the Apple App Store on the mobile devices.

Provisioning is required for connection between DA16200/DA16600 and Router before connecting DA16200/DA16600 with AWS IoT hub. The provisioning can be done with the Renesas Wi-Fi Provisioning app on either an Android or iOS device. For details on how to install and provision the mobile app, see Ref. [4]. When provisioning is completed, select AWS IoT to open AWS application on mobile device.

7.2.1 Open Door

Figure 52 shows message flows of opening the door.



Mobile app:

- Click “ Open door “ button
- Publish a topic

“ thingName/AppControl/0 app_door open “

- Subscribe a topic

“ thingName/DeviceControl/1 mcu_door opened “

- get Shadow
- update UI

DA16200:

- Subscribe a topic

“ thingName/AppControl/0 app_door open “

- send to MCU

"SERVER_DATA 0 app_door open"

- Publish a topic

“ thingName/DeviceControl/1 mcu_door opened “

- send to MCU

" CMD_TO_MCU update "

- update Shadow

"windowStat":"closed","doorStat":opened","temperature":29.000000,"battery":64,"OTAupdate":0,"OTAresult":"OTA_UNKNOWN"

MCU:

- Receive data from DA16200

" SERVER_DATA 0 app_door open "

- Open door
- Send to DA16200

“ MCU_DATA 1 mcu_door opened ”

- Receive data from DA16200

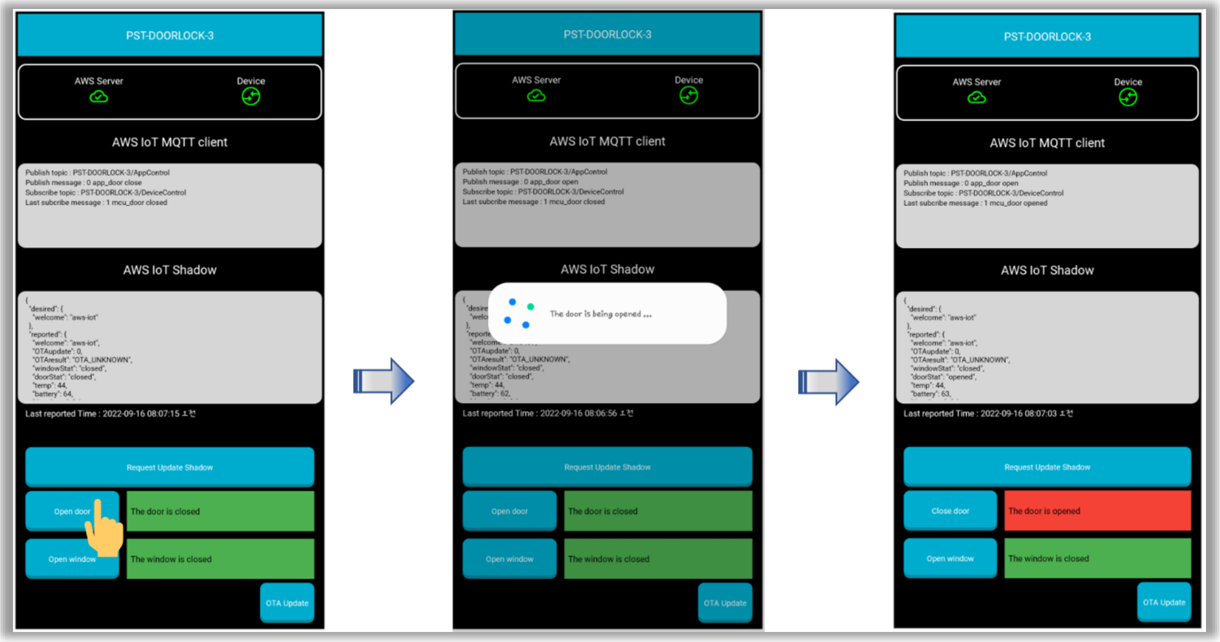
" CMD_TO_MCU update "

- Send to DA16200

“ CMD MCU_DATA 4 battery 64 5 temperature 29 6 doorStat opened 7 windowStat closed”

Figure 52. Message flows of opening door

The operation of **opening door** in Android app is shown in Figure 54.



Publish topic : PST-DOORLOCK-3/AppControl
Publish message : 0 app_door close
Subscribe topic : PST-DOORLOCK-3/DeviceControl
Last subcribe message : 1 mcu_door closed

```
{
  "desired": {
    "welcome": "aws-iot"
  },
  "reported": {
    "welcome": "aws-iot",
    "OTAupdate": 0,
    "OTAresult": "OTA_UNKNOWN",
    "windowStat": "closed",
    "doorStat": "closed",
    "temp": 44,
    "battery": 64,
```

Last reported Time : 2022-09-16 08:07:15 오전

Publish topic : PST-DOORLOCK-3/AppControl
Publish message : 0 app_door open
Subscribe topic : PST-DOORLOCK-3/DeviceControl
Last subcribe message : 1 mcu_door closed

The door is being opened ...

```
    "OTAupdate": 0,
    "OTAresult": "OTA_UNKNOWN",
    "windowStat": "closed",
    "doorStat": "closed",
    "temp": 44,
    "battery": 62,
```

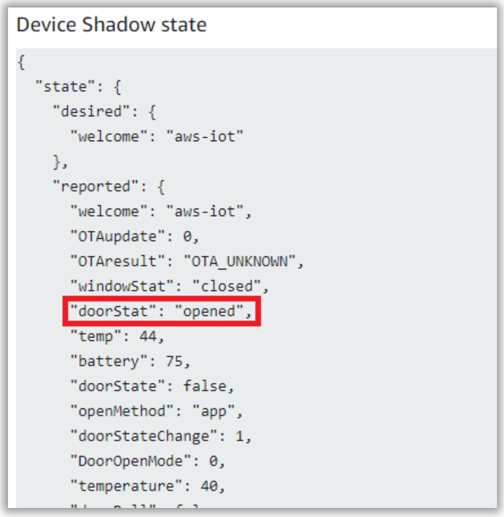
Last reported Time : 2022-09-16 08:06:56 오전

Publish topic : PST-DOORLOCK-3/AppControl
Publish message : 0 app_door open
Subscribe topic : PST-DOORLOCK-3/DeviceControl
Last subcribe message : 1 mcu_door opened

```
{
  "desired": {
    "welcome": "aws-iot"
  },
  "reported": {
    "welcome": "aws-iot",
    "OTAupdate": 0,
    "OTAresult": "OTA_UNKNOWN",
    "windowStat": "closed",
    "doorStat": "opened",
    "temp": 44,
    "battery": 63,
```

Last reported Time : 2022-09-16 08:07:03 오전

Figure 53. Open dooring on mobile app



Device Shadow state

```
{
  "state": {
    "desired": {
      "welcome": "aws-iot"
    },
    "reported": {
      "welcome": "aws-iot",
      "OTAupdate": 0,
      "OTAresult": "OTA_UNKNOWN",
      "windowStat": "closed",
      "doorStat": "opened",
      "temp": 44,
      "battery": 75,
      "doorState": false,
      "openMethod": "app",
      "doorStateChange": 1,
      "DoorOpenMode": 0,
      "temperature": 40,
```

Figure 54. Shadow state when door is open

When the operation of opening door is completed, the console logs of the DA16200 appear as follows:

```
Count : 0, cmdNum = 4

mqtttype = 1

index(=3) matched

data type(shadow) = 0

call update sensor(need to be set variable): battery = 63


Count : 1, cmdNum = 5

mqtttype = 1

index(=2) matched

data type(shadow) = 2

call update sensor(need to be set variable): temperature = 28.000000
```

```
Count : 2, cmdNum = 6
mqtttype = 1
index(=1) matched
data type(shadow) = 1
call update sensor(need to be set variable): doorStat = opened

Count : 3, cmdNum = 7
mqtttype = 1
index(=0) matched
data type(shadow) = 1
call update sensor(need to be set variable): windowStat = closed

release response

*************************************************************************************
publish (shadow sensor update) OK -     payload:
"{"state":{"reported":{"windowStat":"closed","doorStat":"opened","temperature":28.0
00000,"battery":63,"OTAupdate":0,"OTAresult":"OTA_UNKNOWN"}},"clientToken":"PST-DOORLOCK-3-0"}"
```

7.2.2 Close Door

Figure 55 shows message flows of closing door.

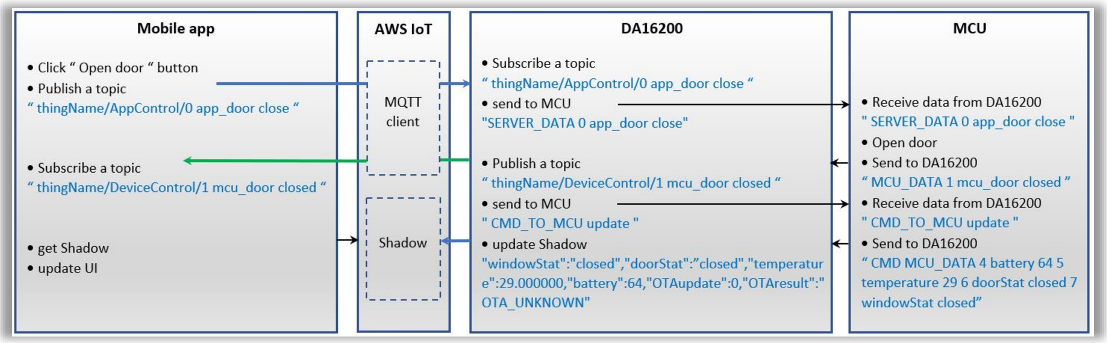

Mobile app
• Click " Open door " button
• Publish a topic
" thingName/AppControl/0 app_door close "
• Subscribe a topic
" thingName/DeviceControl/1 mcu_door closed "
• get Shadow
• update UI
AWS IoT
MQTT client
Shadow
DA16200
• Subscribe a topic
" thingName/AppControl/0 app_door close "
• send to MCU
"SERVER_DATA 0 app_door close"
• Publish a topic
" thingName/DeviceControl/1 mcu_door closed "
• send to MCU
" CMD_TO_MCU update "
• update Shadow
"windowStat":"closed","doorStat":"closed","temperature":29.000000,"battery":64,"OTAupdate":0,"OTAresult":"OTA_UNKNOWN"
MCU
• Receive data from DA16200
" SERVER_DATA 0 app_door close "
• Open door
• Send to DA16200
" MCU_DATA 1 mcu_door closed "
• Receive data from DA16200
" CMD_TO_MCU update "
• Send to DA16200
" CMD MCU_DATA 4 battery 64 5 temperature 29 6 doorStat closed 7 windowStat closed"

Figure 55. Message flows of closing door

The operation of **closing door** in Android app is shown in Figure 56.

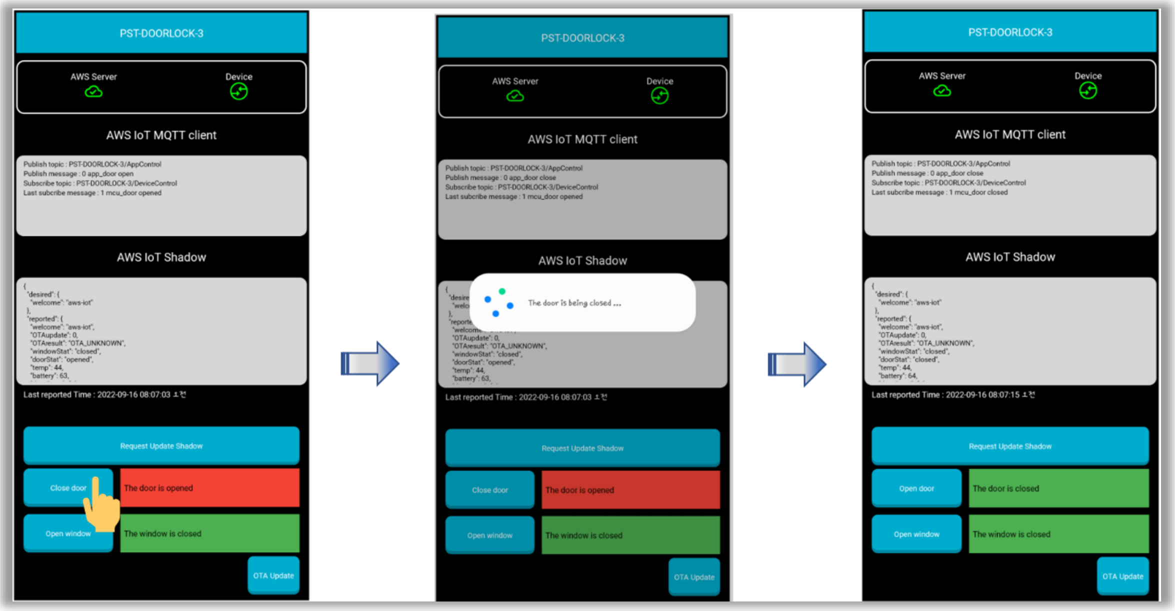
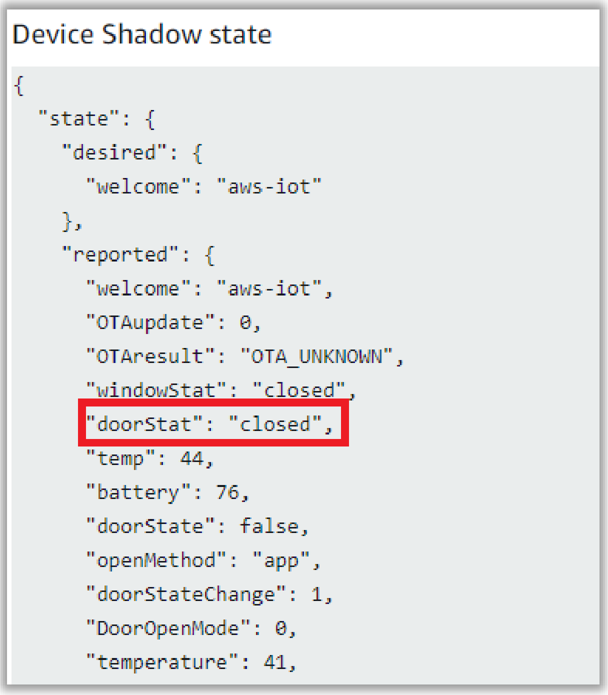


Figure 56. Closing door on mobile app

Figure 57 shows the state of Shadow on the AWS IoT Hub when the operation for closing door is completed.



Device Shadow state

```
{
  "state": {
    "desired": {
      "welcome": "aws-iot"
    },
    "reported": {
      "welcome": "aws-iot",
      "OTAupdate": 0,
      "OTAresult": "OTA_UNKNOWN",
      "windowStat": "closed",
      "doorStat": "closed",
      "temp": 44,
      "battery": 76,
      "doorState": false,
      "openMethod": "app",
      "doorStateChange": 1,
      "DoorOpenMode": 0,
      "temperature": 41,
```

Figure 57. Shadow state when door is closed

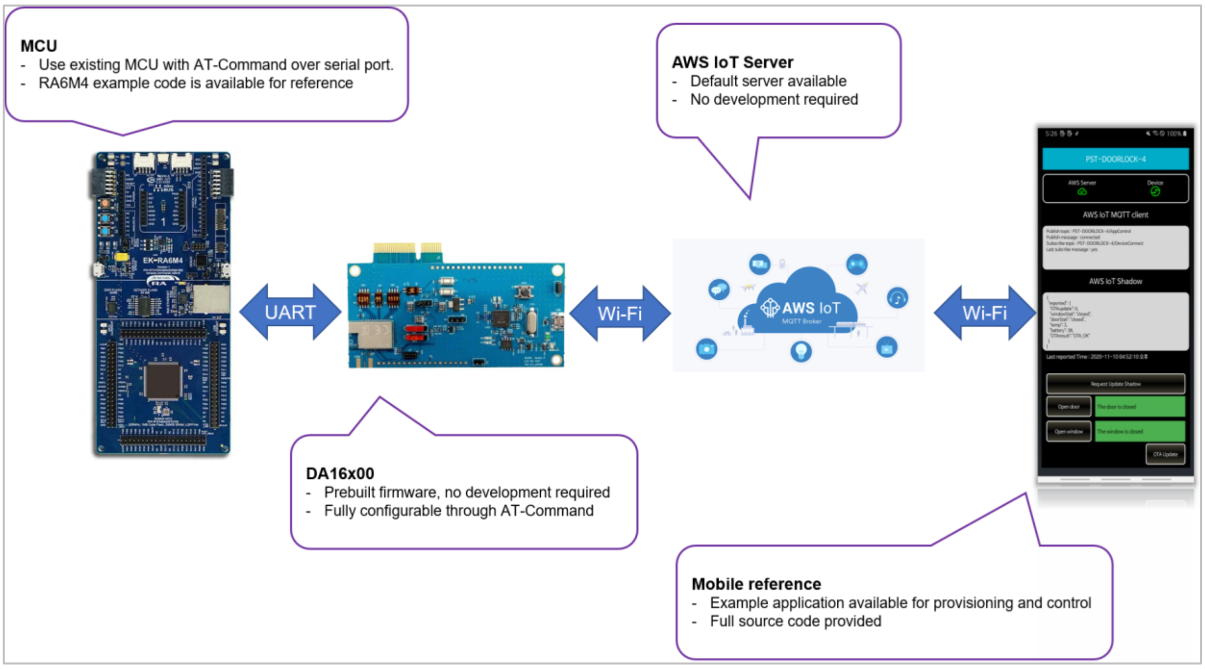
When the operation of closing door is completed, the console logs of the DA16200/DA16600 appears as follows:

```
Count : 0, cmdNum = 4

mqtttype = 1

index(=3) matched

data type(shadow) = 0

call update sensor(need to be set variable): battery = 76
```

```
Count : 1, cmdNum = 5
mqtttype = 1
index(=2) matched
data type(shadow) = 2
call update sensor(need to be set variable): temperature = 41.000000

Count : 2, cmdNum = 6
mqtttype = 1
index(=1) matched
data type(shadow) = 1
call update sensor(need to be set variable): doorStat = closed

Count : 3, cmdNum = 7
mqtttype = 1
index(=0) matched
data type(shadow) = 1
call update sensor(need to be set variable): windowStat = closed

release response

****************************************************************************************
publish (shadow sensor update) OK -     payload:
"{"state":{"reported":{"windowStat":"closed","doorStat":"closed","temperature":41.0
00000,"battery":76,"OTAupdate":0,"OTAresult":"OTA_UNKNOWN"}},"clientToken":"PST-DOORLOCK-3-0"}"
```

7.3 Reference Application in Host MCU

Application in the host MCU can control DA16200/DA16600 and connection between the host MCU and mobile phone through AWS IoT server using AT commands. Figure 58 shows the AWS IoT using firmware images for AT commands and the host MCU.



MCU
- Use existing MCU with AT-Command over serial port.
- RA6M4 example code is available for reference

AWS IoT Server
- Default server available
- No development required

DA16x00
- Prebuilt firmware, no development required
- Fully configurable through AT-Command

Mobile reference
- Example application available for provisioning and control
- Full source code provided

Figure 58. AWS IoT using firmware images for AT commands and host MCU

7.3.1 Download Package for Door Lock Reference Application in Host MCU

A firmware image for AT command and application in MCU are available on the official Renesas website (https://www.renesas.com/us/en/products/wireless-connectivity/wi-fi/low-power-wi-fi).

The contents of the package are the following:

- \DA16200 or \DA16600
  - Firmware images for the DA16200/DA16600 Wi-Fi devices.
  - Tera Term script for downloading the firmware images to the DA16200/DA16600 Wi-Fi device.
- \DA16200\Script (\DA16600\Script)
  - Tera Term script that demonstrates how to use AT commands for AWS IoT using a personal computer and the DA16200/DA16600.
  - Getting Stared with AT commands for AWS IoT
    - Introduces the DA16200/DA16600 AT commands for AWS IoT and describes how to set up the development environment and test the examples.
    - Describes how to connect an external host to the DA16200/DA16600 EVK for using the AT commands for AWS IoT.
    - Describes the AT commands for AWS IoT command list.
- \MCU
  - Sample project based on the RA6M4 development environment which demonstrates how to use AT commands for AWS IoT.

7.3.2 Hardware Connections between DA16200/DA16600 and Host MCU

The hardware components shown in Figure 59 are required to run door lock reference application using AT commands and the host MCU:

- DA16200/DA16600 EVK
- EK-RA6M4 board

- Windows laptop or personal computer.

In addition, the following hardware connections are required for each operation:

- UART0: Programming firmware images and monitoring logs from DA16200/DA16600.
- UART1 or UART2: AT command interface between MCU and DA16200/DA16600.
- GPIO from the MCU to the DA16200/DA16600 to wake up the DA16200/DA16600 from DPM Low-power mode (DPM LPM).
- GPIO from the DA16200/DA16600 to the host MCU to wake up the MCU in Sleep mode.

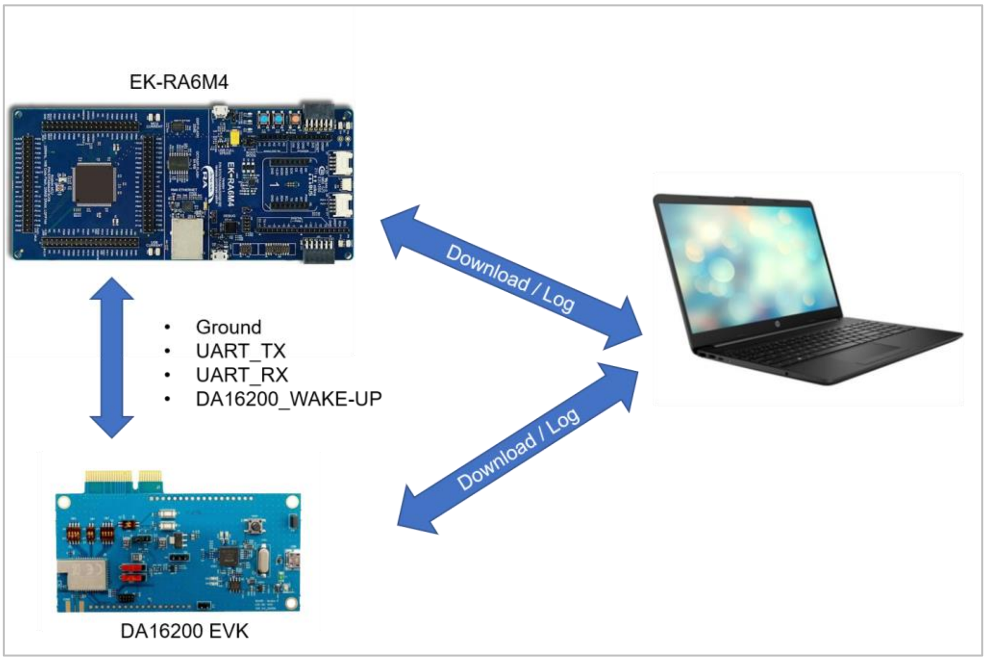


Figure 59. Hardware configuration

Table 1 shows the pin connections between the DA16200/DA16600 EVK and the EK-RA6M4 board.

Table 1. Pin connection

Function	DA16200 EVK		DA16600 EVK		EK-RA6M4 board	
	Pin number	Pin name	Pin number	Pin name	Pin number	Pin name
Ground	J3.18	GND	J2.12	GND	J24-7	GND
UART_TX	J4.11	TX1/GPIOA_4	J2.2	TX2/GPIOC_6	J23-2	D1/TXD
UART_RX	J4.12	RX1/GPIOA_5	J2.4	RX2/GPIOC_7	J23-1	D0/RXD
DA16200_WAKE_UP	J3.11	RTC_WAKE_UP2	SW1	RTC_WAKE_UP2	J23-6	D5/PWM
MCU_WAKE_UP	J4.18	GPIOA_11	J2.9	GPIOA_11	None	None

7.3.2.1 UART Connection for AT Commands

Table 2 shows the default configuration of UART1 (DA16200 EVB) or UART2 (DA16600 EVB) for AT commands.

Table 2. Default configuration for UART1 or UART2

Settings	Value
Baud Rate	115200
Data Bits	8
Parity	None

Settings	Value
Stop Bits	1
Flow Control (Hardware/Software)	None

The DA16200 EVB uses GPIOA_4 and GPIOA_5 for UART1 TX and UART1 RX, and the DA16600 EVB uses GPIOC_6 and GPIOC_7 for UART2 TX and UART2 RX by default. In addition, GND needs to be connected to the host MCU as shown in Figure 60.

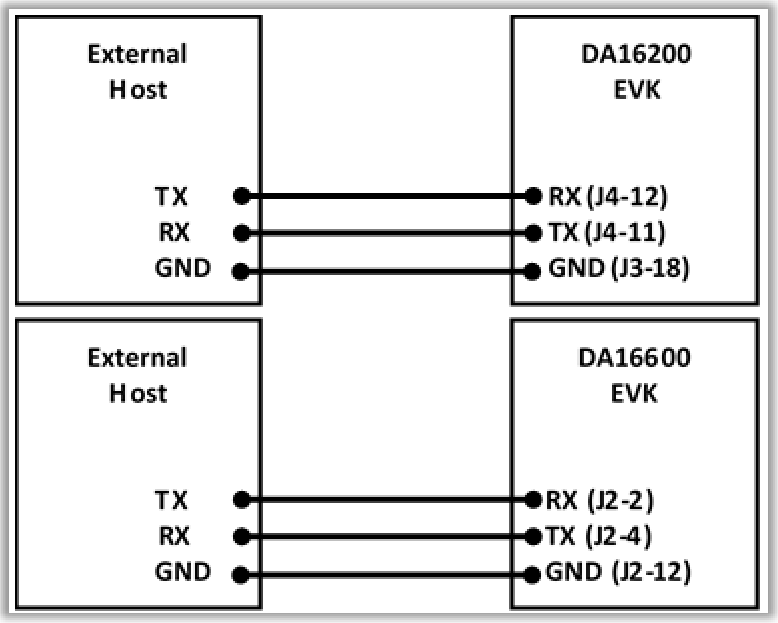


Figure 60. Default UART hardware connection

If the GPIO pin configuration is changed using AT commands, other connections for UART1 can be used as shown in Figure 61. The following AT command is used for GPIOA_2 for UART1 TX and GPIOA_3 for UART1 RX. Table 3 shows the pin combination for UART1.

```
AT+AWS=SET NV_PIN_BMUX BMUX_UART1d              // GPIOA_2 and GPIOA_3 for UART1
```

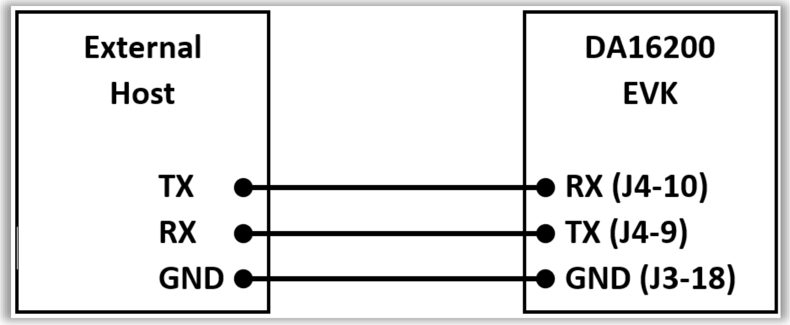


Figure 61. Example of UART1 connection

Table 3. UART1 pin configuration

PIN MUX	GPIO	Signal name
PIN_AMUX	GPIOA_0	TX
	GPIOA_1	RX
PIN_BMUX	GPIOA_2	TX
	GPIOA_3	RX
PIN_CMUX	GPIOA_4	TX

PIN MUX	GPIO	Signal name
	GPIOA_5	RX
PIN_DMUX	GPIOA_6	TX
	GPIOA_7	RX

When Dynamic Power Management (DPM) mode is enabled and DA16200/DA16600 is in DPM LPM, the host MCU must wake up the DA16200/DA16600 from DPM LPM using RTC_WAKE_UP. Then, the host MCU can send or receive data over the network in DPM Fully Functional Mode (FFM). The wake-up event is triggered when the GPIO pin of the host MCU changes from Low to High and then back to Low.

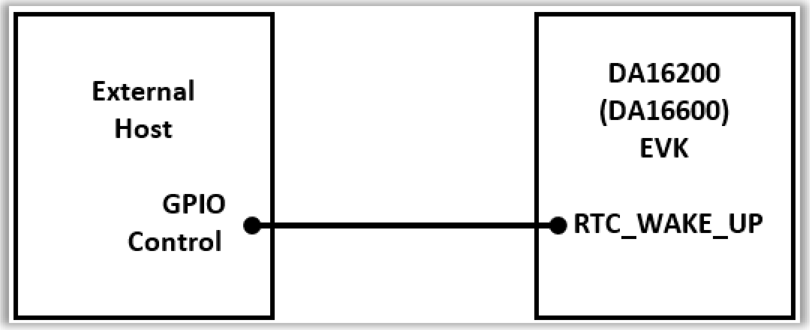


Figure 62. Hardware connection for waking up DA16200/DA16600

The host MCU may be in Sleep mode when DA16200/DA16600 wakes up from DPM LPM and needs to send responses to the host MCU. In this scenario, the DA16200/DA16600 needs to wake up the host MCU from sleep using GPIO as shown in Figure 62. This connection is not required if the host MCU does not use sleep mode. GPIOA_11 is available on DA16200/DA16600 EVB for waking up the host MCU by default (see Figure 63) and it can be configured using the following AT commands:

```
AT+AWS SET APP_MCU_WKAEUP_PORT GPIO_UNIT_A              // GPIO_A port

AT+AWS SET APP_MCU_WKAEUP_PIN GPIO_PIN11         // GPIO_11
```

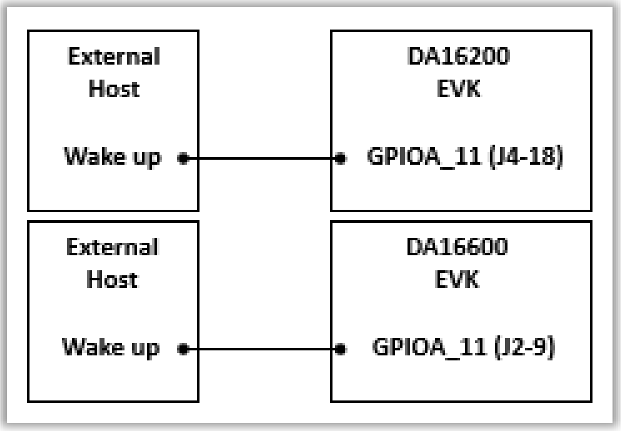


Figure 63. Default pin configuration for waking up host MCU

Other GPIOs in the DA16200 EVB can be used for waking up the host MCU as shown in Table 4. For example, GPIOC_6 can be configured for waking up the host MCU using the following AT commands (see Figure 64):

```
AT+AWS SET APP_MCU_WKAEUP_PORT GPIO_UNIT_C              // GPIO_C port

AT+AWS SET APP_MCU_WKAEUP_PIN GPIO_PIN6         // GPIO_6
```

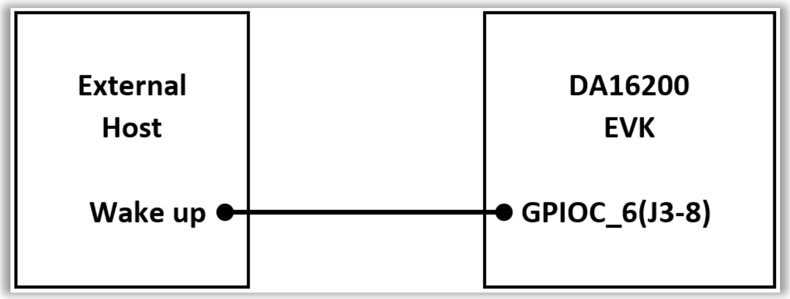


Figure 64. Another pin configuration for waking up host MCU

Table 4. GPIO pin configuration

Port	PIN MUX	GPIO
GPIO_UNIT_A	PIN_AMUX	GPIOA_0
		GPIOA_1
	PIN_BMUX	GPIOA_2
		GPIOA_3
	PIN_CMUX	GPIOA_4
		GPIOA_5
	PIN_DMUX	GPIOA_6
		GPIOA_7
	PIN_EMUX	GPIOA_8
		GPIOA_9
	PIN_FMUX	GPIOA_10
		GPIOA_11
GPIO_UNIT_C	PIN_UMUX	GPIOC_6
		GPIOC_7
		GPIOC_8

7.3.3 Programming Firmware Images for DA16200/DA16600

When using an EVB for the first time, the firmware must be updated to the latest version. For more details, see DA16200 DA16600 FreeRTOS Getting Started Guide, Ref. [3]. After programming the firmware image, factory reset is required to enter the AWS IoT configuration setting mode. This can be done by pushing the "Factory_RST" button for 5 seconds as shown in Figure 65 and Figure 66.

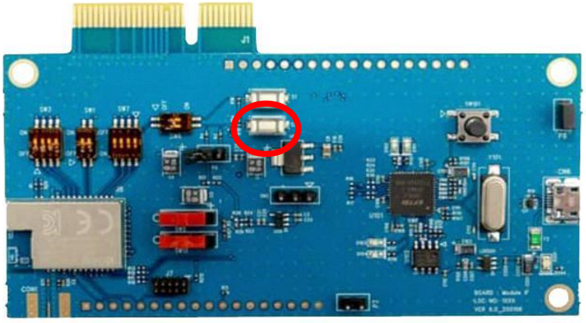


Figure 65. Factory reset button on DA16200 EVB

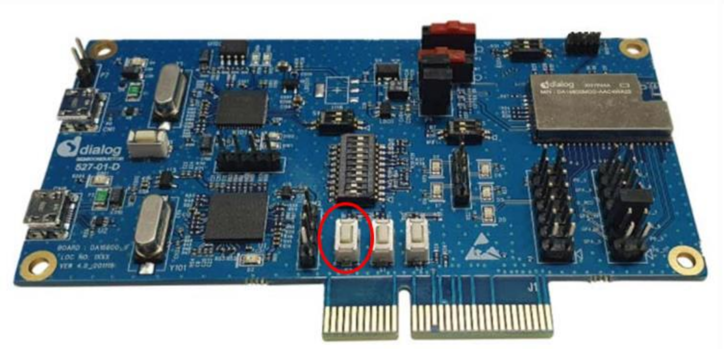


Figure 66. Factory reset button on DA16600 EVB

The logs from DA16200 are as follows:

```
[/DA16200]#
Factory reset ready.


Factory Reseting....


DA16200 concurrent factory reset AP mode = 1 ("AP_ONLY")....


....[app_set_customer_ap_configure] set AP config mode = 0


apps_reboot_ap_mode  Customer configuration ...
.
 default_ssid = "Dialog_DA16200" ..., ap_config_param->ssid_name


 PW = 1234567890


 PW = 1234567890  completed
.
apps_reboot_ap_mode  IPADDR_CUSTOMER...
....
apps_reboot_ap_mode  customer_dhcpd_flag == DHCPD_CUSTOMER..
......
OK
```

The logs from DA16600 are as follows:

```
[/DA16600]#
Factory reset ready.


Factory Reseting....
Set STA Mode ...


Rebooting....


Reset BLE ...


Wakeup source is 0x0
[dpm_init_retmemory] DPM INIT CONFIGURATION(1)


        ******************************************************
        *             DA16600 SDK Information
        * ---------------------------------------------------
        *
        * - CPU Type        : Cortex-M4 (120 MHz)
        * - OS Type          : FreeRTOS 10.4.3
        * - Serial Flash     : 4 MB
        * - SDK Version      : V3.2.8.0 AWS-ATCMD Doorlock Ref. QFN GEN
        * - F/W Version      : FRTOS-GEN01-01-f017bfdf51-006558
        * - F/W Build Time   : Sep  5 2023 17:17:05
        * - Boot Index       : 0
        *
        ******************************************************

gpio wakeup enable 00000402
[combo] dpm_boot_type = 0

>>> UART1 : Clock=80000000, BaudRate=115200
>>> UART1 : DMA Enabled ...
[combo] BLE_BOOT_MODE_0
[combo] BLE FW VER to transfer ....
>>> v_6.0.14.1114.3 (id=1) at bank_1
[combo] BLE FW transfer done

System Mode : Station Only (0)
>>> Start DA16X Supplicant ...
>>> DA16x Supp Ver2.7 - 2022_03
>>> MAC address (sta0) : d4:3d:39:40:72:16
>>> sta0 interface add OK
>>> Start STA mode...
by default, rf_meas_btcoex(1, 0, 0)

>>> UART2 : Clock=80000000, BaudRate=115200
>>> UART2 : DMA Enabled ...
[UART ready notification]
<<< GAPM_DEVICE_READY_IND
AWS IoT dev_name="DA16200", len=7
IoT dev_name="DA16200", len=7
[combo] Advertising...

[/DA16600] #
```

After the factory reset, the DA16200/DA16600 is now ready to enter the AWS IoT Configuration Settings.

7.3.4 Configure Components for Testing

The following information are required for testing the application with AWS IoT server:

- Unique thing name

The information can be set in the source code for the host MCU or using the provided scripts in the downloaded package. For how to run the macro script, see Ref. [3]. The scripts are in the following location:

\DA16x00_img\script\doorlock.ttl

```
;In order to use this script on DA16200, the console should be prompt

;after setting the DA16200 to STA mode, SNTP client enable, and no DPM mode in easy setup through the console.

;set configurations with DA16200's console


;set features

sendln "user"

;set board type

sendln "SET APP_BOARD_FEATURE EVK"

mpause 400

;set your thingname

;sendln "SET APP_THINGNAME FAE-DOORLOCK-4”

mpause 400

;set broker address

sendln "SET AWS_BROKER a1kzdt4nun8bnh-ats.iot.ap-northeast-2.amazonaws.com"

mpause 400
```

The MCU source code can be found in the following file:

\MCU\RA6M4\Src\atcmd\at_cmd.c.

```
#define MAX_RETRY_SEND_COUNT       10


/* AWS features, configurations, and certification keys */

const char* cmd_set_cfg[MAX_CFG_NUM] =

{

  "\r\nAT+"PLATFORM" SET AWS_USE_FP 0\r\n",

  "\r\nAT+"PLATFORM" SET APP_BOARD_FEATURE EVK\r\n",

  "\r\nAT+"PLATFORM" SET APP_THINGNAME FAE-DOORLOCK-4\r\n",

  "\r\nAT+"PLATFORM" SET AWS_BROKER a1kzdt4nun8bnh-ats.iot.ap-northeast-2.amazonaws.com\r\n",

  "\r\nAT+"PLATFORM" SET APP_LPORT 1883\r\n",

  "\r\nAT+"PLATFORM" SET APP_SUBTOPIC /AppControl\r\n",

  "\r\nAT+"PLATFORM" SET APP_PUBTOPIC /DeviceControl\r\n",

  "\r\nAT+"PLATFORM" CFG 0 app_door 1 2\r\n",             /* mcu sub.       str */

  "\r\nAT+"PLATFORM" CFG 1 mcu_door 1 0\r\n",             /* mcu pub.       str */

  "\r\nAT+"PLATFORM" CFG 2 app_window 1 2\r\n",           /* mcu sub.       str */

  "\r\nAT+"PLATFORM" CFG 3 mcu_window 1 0\r\n",           /* mcu pub.       str */

  "\r\nAT+"PLATFORM" CFG 4 battery 0 1\r\n",              /* shadow   int */

  "\r\nAT+"PLATFORM" CFG 5 temperature 2 1\r\n",          /* shadow   float */
  "\r\nAT+"PLATFORM" CFG 6 doorStat 1 1\r\n",            /* shadow  str */
  "\r\nAT+"PLATFORM" CFG 7 windowStat 1 1\r\n",          /* shadow  str */
  "\r\nAT+"PLATFORM" CFG 8 app_shadow 1 2\r\n",          /* mcu sub.       str */
  "\r\nAT+"PLATFORM" CFG 9 mcu_shadow 1 0\r\n",          /* mcu pub.       str */
  "\r\nAT+"PLATFORM" SET SLEEP_MODE 3\r\n",
  "\r\nAT+"PLATFORM" SET USE_DPM 1\r\n",
  "\r\nAT+"PLATFORM" SET RTC_TIME 1740\r\n",
  "\r\nAT+"PLATFORM" SET DPM_KEEP_ALIVE 30000\r\n",
  "\r\nAT+"PLATFORM" SET USE_WAKE_UP 0\r\n",
  "\r\nAT+"PLATFORM" SET TIM_WAKE_UP 10\r\n",
  "\r\nAT+"PLATFORM" SET APP_MCU_WKAEUP_PORT GPIO_UNIT_A\r\n",    /* GPIO_UNIT_A or
GPIO_UNIT_C */
  "\r\nAT+"PLATFORM" SET APP_MCU_WKAEUP_PIN GPIO_PIN11\r\n"       /* GPIO_PIN0 ~ GPIO_PIN11
or GPIO_PIN6~GPIO_PIN8 */
};
```

7.3.5 Test without Host MCU

If the host MCU is not available, the AWS IoT commands can be tested with the script provided in the downloaded package.

Door lock for two-way communication:

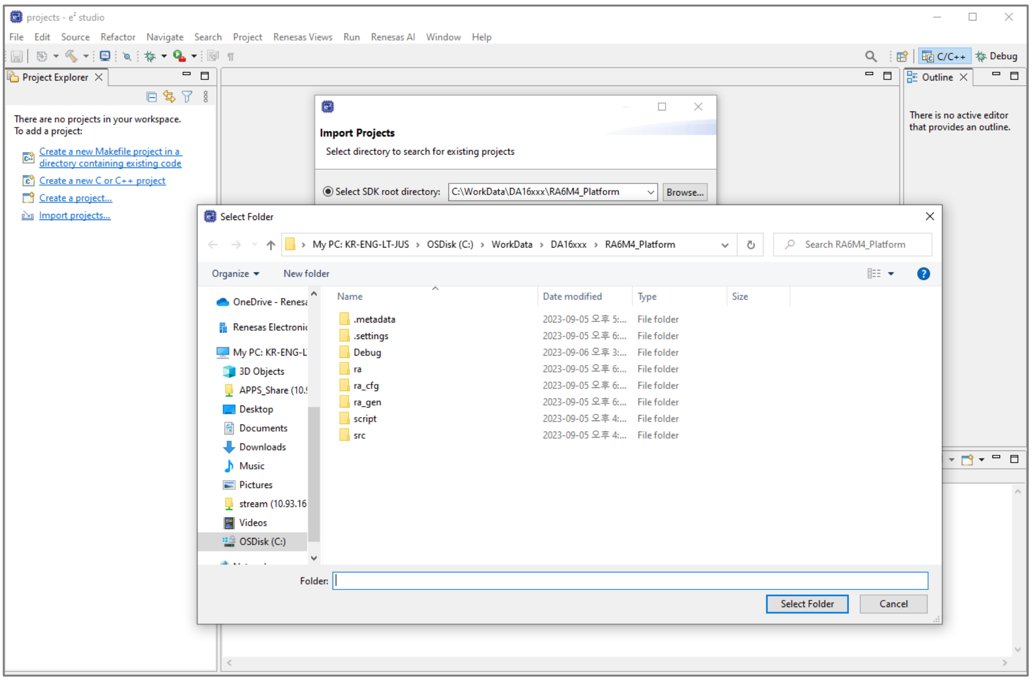
`\DA16x00_img\script\doorlock.ttl`.

NOTE
The example script only supports initial value setting. To fully verify the operation of the AT commands, use the host MCU for interacting with the server and application.

7.3.6 Test with Host MCU

The e^2studio is required for building source code for the host MCU and programing the images to the host MCU. Visit the Renesas website (https://www.renesas.com/us/en/software-tool/e-studio) for downloading and installing the e^2studio. After installing the e^2studio, complete the following steps for building and programming.

1. Import the `project` file to `\MCU\RA6M4\`.


projects - e² studio
File Edit Source Refactor Navigate Search Project Renesas Views Run Renesas AI Window Help
Project Explorer
Outline
C/C++
Debug
There are no projects in your workspace.
To add a project:
Create a new Makefile project in a directory containing existing code
Create a new C or C++ project
Create a project...
Import projects...
There is no active editor that provides an outline.
Import Projects
Select directory to search for existing projects
Browse...
Select Folder
My PC: KR-ENG-LT-JUS › OSDisk (C:) › WorkData › DA16xxx › RA6M4_Platform
Search RA6M4_Platform
Organize
New folder
Name
Date modified
Type
Size
OneDrive - Renesa
Renesas Electronic
My PC: KR-ENG-L
3D Objects
APPS_Share (10.
Desktop
Documents
Downloads
Music
Pictures
stream (10.93.16
Videos
OSDisk (C:)
Select Folder
Cancel

Select SDK root directory: C:\WorkData\DA16xxx\RA6M4_Platform

Name	Date modified	Type
.metadata	2023-09-05 오후 5:...	File folder
.settings	2023-09-05 오후 6:...	File folder
Debug	2023-09-06 오후 3:...	File folder
ra	2023-09-05 오후 6:...	File folder
ra_cfg	2023-09-05 오후 6:...	File folder
ra_gen	2023-09-05 오후 6:...	File folder
script	2023-09-05 오후 4:...	File folder
src	2023-09-05 오후 4:...	File folder

Folder:

Figure 67. e^{2}studio project file

NOTE
When connecting to the RA6M4 MCU for the first time or changing the configuration, complete the step 2 to set up the FSP configuration.

2. To set FSP configuration of the RA6M4 MCU, select **configurations.xml**.

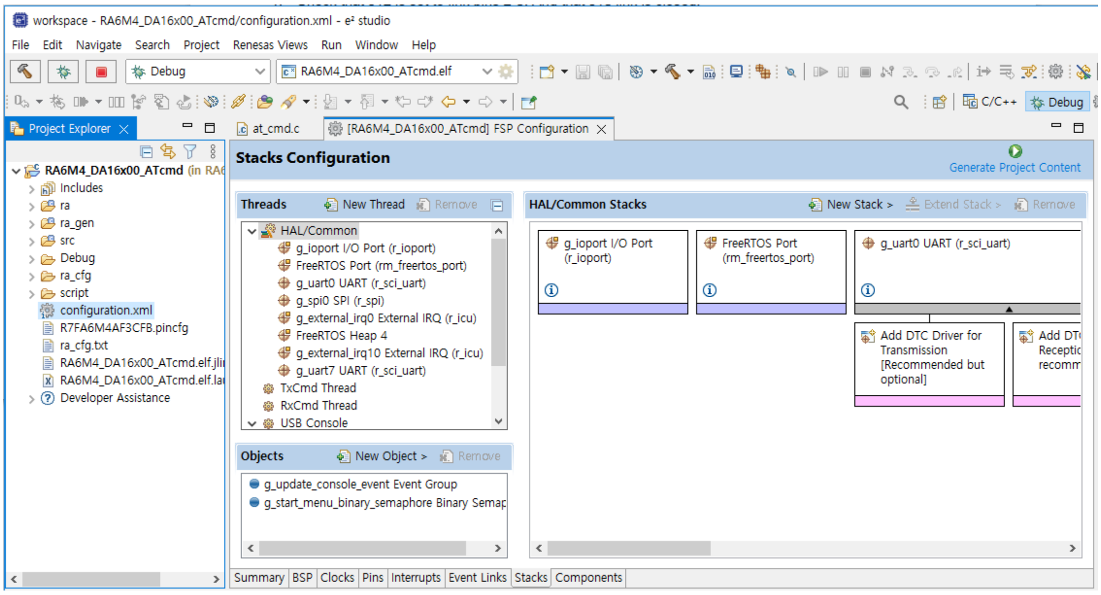
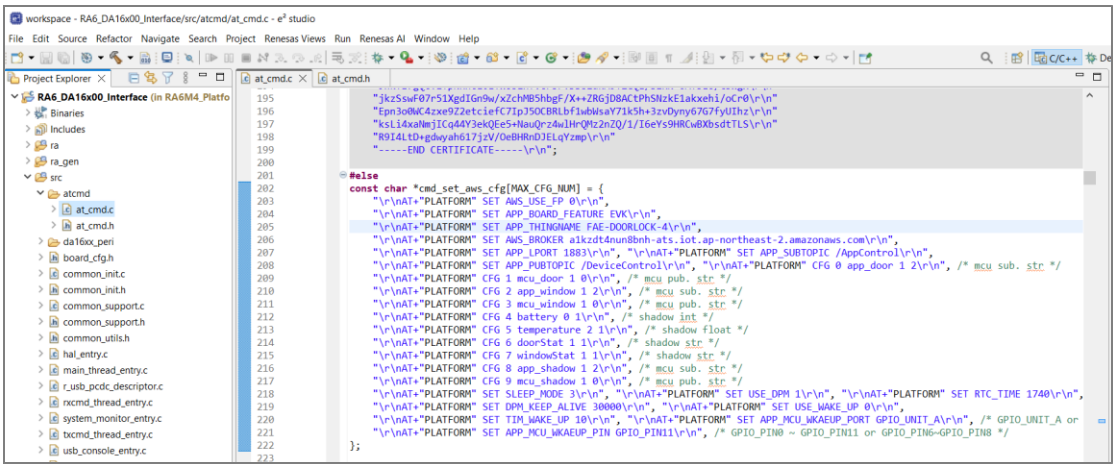

workspace - RA6M4_DA16x00_ATcmd/configuration.xml - e² studio
File Edit Navigate Search Project Renesas Views Run Window Help
Debug
RA6M4_DA16x00_ATcmd.elf
C/C++
Debug
Project Explorer
RA6M4_DA16x00_ATcmd (in RA6
Includes
ra
ra_gen
src
Debug
ra_cfg
script
configuration.xml
R7FA6M4AF3CFB.pincfg
ra_cfg.txt
RA6M4_DA16x00_ATcmd.elf.jli
RA6M4_DA16x00_ATcmd.elf.la
Developer Assistance
at_cmd.c
[RA6M4_DA16x00_ATcmd] FSP Configuration
Stacks Configuration
Generate Project Content
Threads
New Thread
Remove
HAL/Common
g_ioport I/O Port (r_ioport)
FreeRTOS Port (rm_freertos_port)
g_uart0 UART (r_sci_uart)
g_spi0 SPI (r_spi)
g_external_irq0 External IRQ (r_icu)
FreeRTOS Heap 4
g_external_irq10 External IRQ (r_icu)
g_uart7 UART (r_sci_uart)
TxCmd Thread
RxCmd Thread
USB Console
HAL/Common Stacks
New Stack >
Extend Stack >
Remove
g_ioport I/O Port (r_ioport)
FreeRTOS Port (rm_freertos_port)
g_uart0 UART (r_sci_uart)
Add DTC Driver for Transmission [Recommended but optional]
Add DTC Receptio recomm
Objects
New Object >
Remove
g_update_console_event Event Group
g_start_menu_binary_semaphore Binary Semap
Summary BSP Clocks Pins Interrupts Event Links Stacks Components

Figure 68. FSP configuration

3. Use the thing name received from the FAE to test without setting up a server.

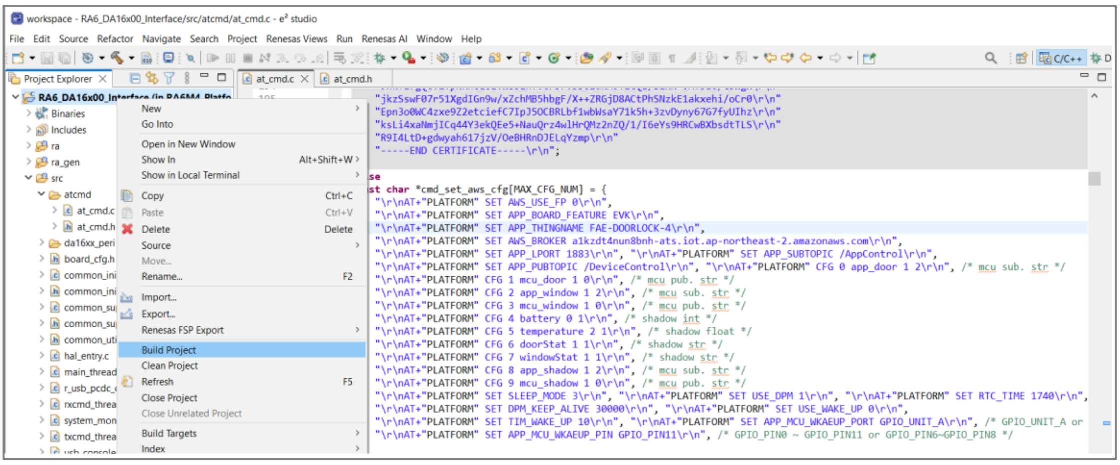
4. Change the thing name to the received name.



```
    "jkzSswF07r51XgdIGn9w/xZchMB5hbgF/X++ZRGjD8ACtPhSNzkE1akxehi/oCr0\r\n"
    "Epn3o0WC4zxe9Z2etciefC7IpJ5OCBRLbf1wbWsaY71k5h+3zvDyny67G7fyUIhz\r\n"
    "ksLi4xaNmjICq44Y3ekQEe5+NauQrz4wlHrQMz2nZQ/1/I6eYs9HRCwBXbsdtTLS\r\n"
    "R9I4LtD+gdwyah617jzV/OeBHRnDJELqYzmp\r\n"
    "-----END CERTIFICATE-----\r\n";

#else
const char *cmd_set_aws_cfg[MAX_CFG_NUM] = {
    "\r\nAT+"PLATFORM" SET AWS_USE_FP 0\r\n",
    "\r\nAT+"PLATFORM" SET APP_BOARD_FEATURE EVK\r\n",
    "\r\nAT+"PLATFORM" SET APP_THINGNAME FAE-DOORLOCK-4\r\n",
    "\r\nAT+"PLATFORM" SET AWS_BROKER a1kzdt4nun8bnh-ats.iot.ap-northeast-2.amazonaws.com\r\n",
    "\r\nAT+"PLATFORM" SET APP_LPORT 1883\r\n", "\r\nAT+"PLATFORM" SET APP_SUBTOPIC /AppControl\r\n",
    "\r\nAT+"PLATFORM" SET APP_PUBTOPIC /DeviceControl\r\n", "\r\nAT+"PLATFORM" CFG 0 app_door 1 2\r\n", /* mcu sub. str */
    "\r\nAT+"PLATFORM" CFG 1 mcu_door 1 0\r\n", /* mcu pub. str */
    "\r\nAT+"PLATFORM" CFG 2 app_window 1 2\r\n", /* mcu sub. str */
    "\r\nAT+"PLATFORM" CFG 3 mcu_window 1 0\r\n", /* mcu pub. str */
    "\r\nAT+"PLATFORM" CFG 4 battery 0 1\r\n", /* shadow int */
    "\r\nAT+"PLATFORM" CFG 5 temperature 2 1\r\n", /* shadow float */
    "\r\nAT+"PLATFORM" CFG 6 doorStat 1 1\r\n", /* shadow str */
    "\r\nAT+"PLATFORM" CFG 7 windowStat 1 1\r\n", /* shadow str */
    "\r\nAT+"PLATFORM" CFG 8 app_shadow 1 2\r\n", /* mcu sub. str */
    "\r\nAT+"PLATFORM" CFG 9 mcu_shadow 1 0\r\n", /* mcu pub. str */
    "\r\nAT+"PLATFORM" SET SLEEP_MODE 3\r\n", "\r\nAT+"PLATFORM" SET USE_DPM 1\r\n", "\r\nAT+"PLATFORM" SET RTC_TIME 1740\r\n",
    "\r\nAT+"PLATFORM" SET DPM_KEEP_ALIVE 30000\r\n", "\r\nAT+"PLATFORM" SET USE_WAKE_UP 0\r\n",
    "\r\nAT+"PLATFORM" SET TIM_WAKE_UP 10\r\n", "\r\nAT+"PLATFORM" SET APP_MCU_WKAEUP_PORT GPIO_UNIT_A\r\n", /* GPIO_UNIT_A or
    "\r\nAT+"PLATFORM" SET APP_MCU_WKAEUP_PIN GPIO_PIN11\r\n", /* GPIO_PIN0 ~ GPIO_PIN11 or GPIO_PIN6~GPIO_PIN8 */
};
```

Figure 69. Thing name in MCU source code

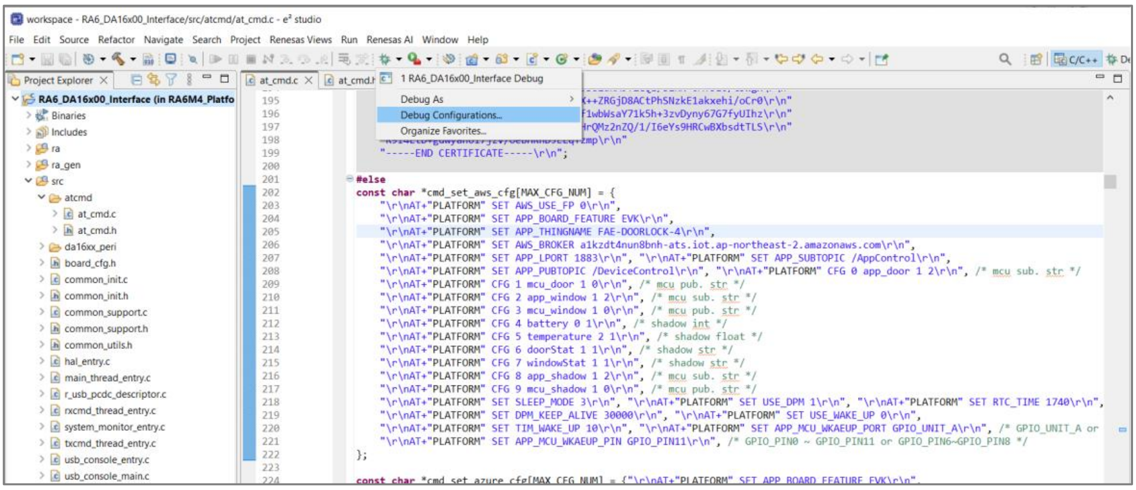
5. To build a new project, select **project > Build Project**.



```
    "jkzSswF07r51XgdIGn9w/xZchMB5hbgF/X++ZRGjD8ACtPhSNzkE1akxehi/oCr0\r\n"
    "Epn3o0WC4zxe9Z2etciefC7IpJ5OCBRLbf1wbWsaY71k5h+3zvDyny67G7fyUIhz\r\n"
    "ksLi4xaNmjICq44Y3ekQEe5+NauQrz4wlHrQMz2nZQ/1/I6eYs9HRCwBXbsdtTLS\r\n"
    "R9I4LtD+gdwyah617jzV/OeBHRnDJELqYzmp\r\n"
    "-----END CERTIFICATE-----\r\n";

se
st char *cmd_set_aws_cfg[MAX_CFG_NUM] = {
    "\r\nAT+"PLATFORM" SET AWS_USE_FP 0\r\n",
    "\r\nAT+"PLATFORM" SET APP_BOARD_FEATURE EVK\r\n",
    "\r\nAT+"PLATFORM" SET APP_THINGNAME FAE-DOORLOCK-4\r\n",
    "\r\nAT+"PLATFORM" SET AWS_BROKER a1kzdt4nun8bnh-ats.iot.ap-northeast-2.amazonaws.com\r\n",
    "\r\nAT+"PLATFORM" SET APP_LPORT 1883\r\n", "\r\nAT+"PLATFORM" SET APP_SUBTOPIC /AppControl\r\n",
    "\r\nAT+"PLATFORM" SET APP_PUBTOPIC /DeviceControl\r\n", "\r\nAT+"PLATFORM" CFG 0 app_door 1 2\r\n", /* mcu sub. str */
    "\r\nAT+"PLATFORM" CFG 1 mcu_door 1 0\r\n", /* mcu pub. str */
    "\r\nAT+"PLATFORM" CFG 2 app_window 1 2\r\n", /* mcu sub. str */
    "\r\nAT+"PLATFORM" CFG 3 mcu_window 1 0\r\n", /* mcu pub. str */
    "\r\nAT+"PLATFORM" CFG 4 battery 0 1\r\n", /* shadow int */
    "\r\nAT+"PLATFORM" CFG 5 temperature 2 1\r\n", /* shadow float */
    "\r\nAT+"PLATFORM" CFG 6 doorStat 1 1\r\n", /* shadow str */
    "\r\nAT+"PLATFORM" CFG 7 windowStat 1 1\r\n", /* shadow str */
    "\r\nAT+"PLATFORM" CFG 8 app_shadow 1 2\r\n", /* mcu sub. str */
    "\r\nAT+"PLATFORM" CFG 9 mcu_shadow 1 0\r\n", /* mcu pub. str */
    "\r\nAT+"PLATFORM" SET SLEEP_MODE 3\r\n", "\r\nAT+"PLATFORM" SET USE_DPM 1\r\n", "\r\nAT+"PLATFORM" SET RTC_TIME 1740\r\n",
    "\r\nAT+"PLATFORM" SET DPM_KEEP_ALIVE 30000\r\n", "\r\nAT+"PLATFORM" SET USE_WAKE_UP 0\r\n",
    "\r\nAT+"PLATFORM" SET TIM_WAKE_UP 10\r\n", "\r\nAT+"PLATFORM" SET APP_MCU_WKAEUP_PORT GPIO_UNIT_A\r\n", /* GPIO_UNIT_A or
    "\r\nAT+"PLATFORM" SET APP_MCU_WKAEUP_PIN GPIO_PIN11\r\n", /* GPIO_PIN0 ~ GPIO_PIN11 or GPIO_PIN6~GPIO_PIN8 */
```

Figure 70. Build project

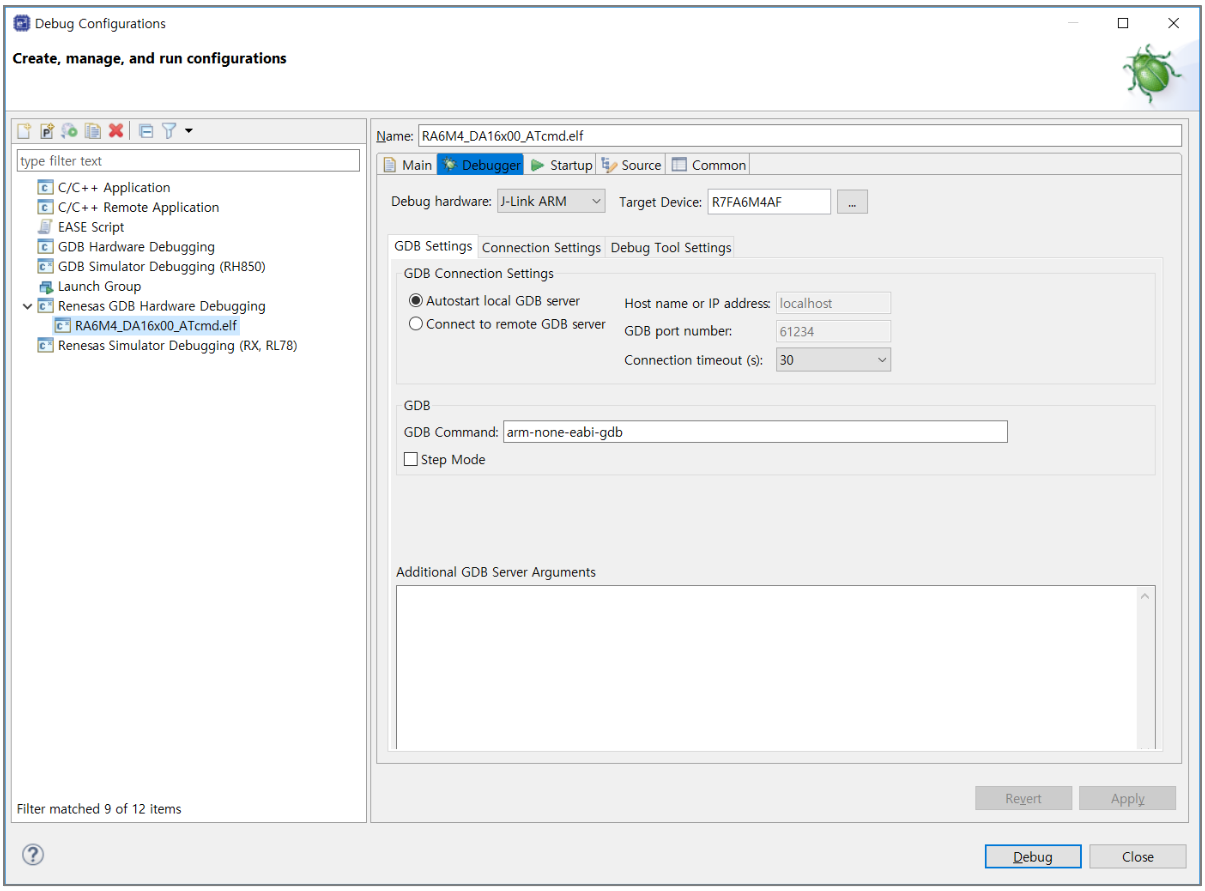
6. To set the connection to the RA6M4 MCU, select **Debug Configurations**.



```
                                    X++ZRGjD8ACtPhSNzkE1akxehi/oCr0\r\n"
                                    f1wbWsaY71k5h+3zvDyny67G7fyUIhz\r\n"
                                    HrQMz2nZQ/1/I6eYs9HRCwBXbsdtTLS\r\n"
                                    mp\r\n"
    "-----END CERTIFICATE-----\r\n";

#else
const char *cmd_set_aws_cfg[MAX_CFG_NUM] = {
    "\r\nAT+"PLATFORM" SET AWS_USE_FP 0\r\n",
    "\r\nAT+"PLATFORM" SET APP_BOARD_FEATURE EVK\r\n",
    "\r\nAT+"PLATFORM" SET APP_THINGNAME FAE-DOORLOCK-4\r\n",
    "\r\nAT+"PLATFORM" SET AWS_BROKER a1kzdt4nun8bnh-ats.iot.ap-northeast-2.amazonaws.com\r\n",
    "\r\nAT+"PLATFORM" SET APP_LPORT 1883\r\n", "\r\nAT+"PLATFORM" SET APP_SUBTOPIC /AppControl\r\n",
    "\r\nAT+"PLATFORM" SET APP_PUBTOPIC /DeviceControl\r\n", "\r\nAT+"PLATFORM" CFG 0 app_door 1 2\r\n", /* mcu sub. str */
    "\r\nAT+"PLATFORM" CFG 1 mcu_door 1 0\r\n", /* mcu pub. str */
    "\r\nAT+"PLATFORM" CFG 2 app_window 1 2\r\n", /* mcu sub. str */
    "\r\nAT+"PLATFORM" CFG 3 mcu_window 1 0\r\n", /* mcu pub. str */
    "\r\nAT+"PLATFORM" CFG 4 battery 0 1\r\n", /* shadow int */
    "\r\nAT+"PLATFORM" CFG 5 temperature 2 1\r\n", /* shadow float */
    "\r\nAT+"PLATFORM" CFG 6 doorStat 1 1\r\n", /* shadow str */
    "\r\nAT+"PLATFORM" CFG 7 windowStat 1 1\r\n", /* shadow str */
    "\r\nAT+"PLATFORM" CFG 8 app_shadow 1 2\r\n", /* mcu sub. str */
    "\r\nAT+"PLATFORM" CFG 9 mcu_shadow 1 0\r\n", /* mcu pub. str */
    "\r\nAT+"PLATFORM" SET SLEEP_MODE 3\r\n", "\r\nAT+"PLATFORM" SET USE_DPM 1\r\n", "\r\nAT+"PLATFORM" SET RTC_TIME 1740\r\n",
    "\r\nAT+"PLATFORM" SET DPM_KEEP_ALIVE 30000\r\n", "\r\nAT+"PLATFORM" SET USE_WAKE_UP 0\r\n",
    "\r\nAT+"PLATFORM" SET TIM_WAKE_UP 10\r\n", "\r\nAT+"PLATFORM" SET APP_MCU_WKAEUP_PORT GPIO_UNIT_A\r\n", /* GPIO_UNIT_A or
    "\r\nAT+"PLATFORM" SET APP_MCU_WKAEUP_PIN GPIO_PIN11\r\n", /* GPIO_PIN0 ~ GPIO_PIN11 or GPIO_PIN6~GPIO_PIN8 */
};

const char *cmd_set_azure_cfg[MAX_CFG_NUM] = {"\r\nAT+"PLATFORM" SET APP_BOARD_FEATURE EVK\r\n",
```

Figure 71. Debug configurations

7. On the **Debugger** tab, change the configuration as shown in Figure 72, and then click **Apply** > **Debug**.



Debug Configurations

Create, manage, and run configurations

type filter text

- C/C++ Application
- C/C++ Remote Application
- EASE Script
- GDB Hardware Debugging
- GDB Simulator Debugging (RH850)
- Launch Group
- Renesas GDB Hardware Debugging
  - RA6M4_DA16x00_ATcmd.elf
- Renesas Simulator Debugging (RX, RL78)

Filter matched 9 of 12 items

Name: RA6M4_DA16x00_ATcmd.elf

Main | Debugger | Startup | Source | Common

Debug hardware: J-Link ARM Target Device: R7FA6M4AF

GDB Settings | Connection Settings | Debug Tool Settings

GDB Connection Settings

● Autostart local GDB server
○ Connect to remote GDB server

Host name or IP address: localhost
GDB port number: 61234
Connection timeout (s): 30

GDB

GDB Command: arm-none-eabi-gdb

□ Step Mode

Additional GDB Server Arguments

Figure 72. Set debug configurations

The following shows the console output of the DA16200 after a factory mode reset.

```
Soft AP is Ready (d4:3d:39:10:d5:07)


>>> UART1 : Clock=80000000, BaudRate=115200

>>> UART1 : DMA Enabled ...

[UART ready notification]

[http_server_task] HTTP-Server Start!!

=============================================

[ AWS-IOT AT COMMAND ]

[ aws_shadow_dpm_auto_start]

AWS_IOT on Station Mode for "FAE-DOORLOCK-4"

==============================================

[pal_app_dpm_auto_start] mcu_wakeup_port=-1, mcu_wakeup_pin=0x0

default set to mcu_wakeup_port=0, mcu_wakeup_pin=0x800


Root CA: X

Certificate: X

Private Key: X
```

```
nvram read string(thingname) error
invalid APP feature...can't start APP Platform thread...check again
.. UART ready
```

The following shows the console output of the DA16200 when setting the AWS IoT configuration with AT commands from an MCU.

```
=======================================================
argc num = 2
argv[0]: AT+AWS
argv[1]: CFG 3 mcu_window 1 0
=======================================================


=======================================================
Att[3] number   : 3
Att[3] name       : mcu_window
Att[3] data type: 1
Att[3] MQTT type: 0
=======================================================


=======================================================
argc num = 2
argv[0]: AT+AWS
argv[1]: CFG 4 battery 0 1
=======================================================


=======================================================
Att[4] number   : 4
Att[4] name       : battery
Att[4] data type: 0
Att[4] MQTT type: 1
=======================================================



=======================================================
argc num = 2
argv[0]: AT+AWS
argv[1]: CFG 5 temperature 2 1
=======================================================


=======================================================
Att[5] number   : 5
Att[5] name       : temperature
Att[5] data type: 2
```

```
Att[5] MQTT type: 1
=======================================================
```

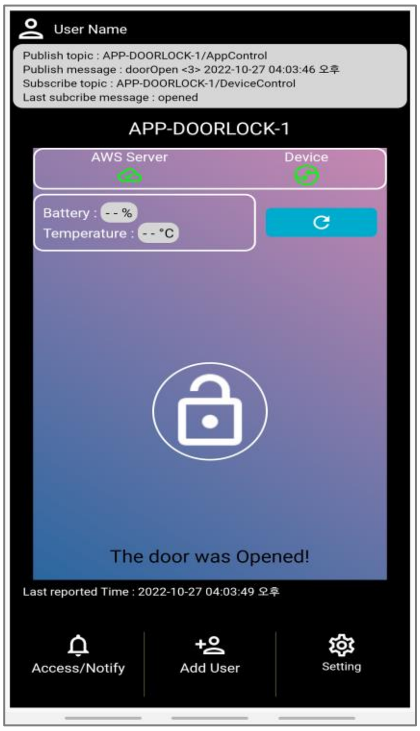
The following shows the console output of the DA16200 after the Soft AP was configured and it is waiting to be provisioned by the mobile application.

```
Soft AP is Ready (d4:3d:39:10:d5:07)

>>> UART1 : Clock=80000000, BaudRate=115200
>>> UART1 : DMA Enabled ...
[UART ready notification]
[http_server_task] HTTP-Server Start!!
===========================================
[ AWS-IOT AT COMMAND ]
[ aws_shadow_dpm_auto_start]
AWS_IOT on Station Mode for "FAE-DOORLOCK-4"
============================================
[pal_app_dpm_auto_start] mcu_wakeup_port=0, mcu_wakeup_pin=0x800

Root CA: O
Certificate: O
Private Key: O

subscribe index=0, name=app_door
subscribe index=2, name=app_window
newNode index=4
newNode index=5
newNode index=6
newNode index=7
subscribe index=8, name=app_shadow
shadow item count = 4, (integer#=1, string#=2, float#=1)
current shadowConut = 4
pkey=windowStat, pdata=test
current shadowConut = 3
pkey=doorStat, pdata=test
current shadowConut = 2
pkey=temperature, pdata=16.500000
current shadowConut = 1
pkey=battery, pdata=2700

AWS_IOT AP Mode  FAE-DOORLOCK-4

+ATPROV=STATUS 1
=======================================================
[Start Provisioning with TCP/TLS] .. Soft AP Mode
=======================================================
[app_provision_switch_client_thread] Create...(status=0) [10]
[app_provision_TCP_server_thread] Create ...
[app_provision_TLS_server_thread] Create TLS...

>>> Start Provisioning Server (TLS) ...
Wait Accept (TLS)...
[app_find_home_ap] Wi-Fi Scan request success.
[app_find_home_ap:518] (0) iptime_justin / 3 / -34 / 2447
[app_find_home_ap:518] (1) AP-101-201 / 3 / -66 / 2432
[app_find_home_ap:518] (2) SK_WiFiGIGA551A_2.4G / 3 / -78 / 2422
[app_find_home_ap:518] (3) SK_WiFiGIGA551A / 3 / -79 / 2422
[app_find_home_ap:518] (4) SK_WiFi3801 / 3 / -94 / 2412
[app_find_home_ap:518] (5) NIS-HomeAP11N / 0 / -74 / 2447
[app_provision_TCP_server_thread] socket().. status=1
 Wait Accept...
```

7.4 Mobile App Demo

Install the mobile application by searching for **DA16200** or **DA16600** in the Google Pay Store or the Apple App Store on the mobile devices.

7.4.1 Open Door



User Name

Publish topic : APP-DOORLOCK-1/AppControl
Publish message : doorOpen <3> 2022-10-27 04:03:46 오후
Subscribe topic : APP-DOORLOCK-1/DeviceControl
Last subcribe message : opened

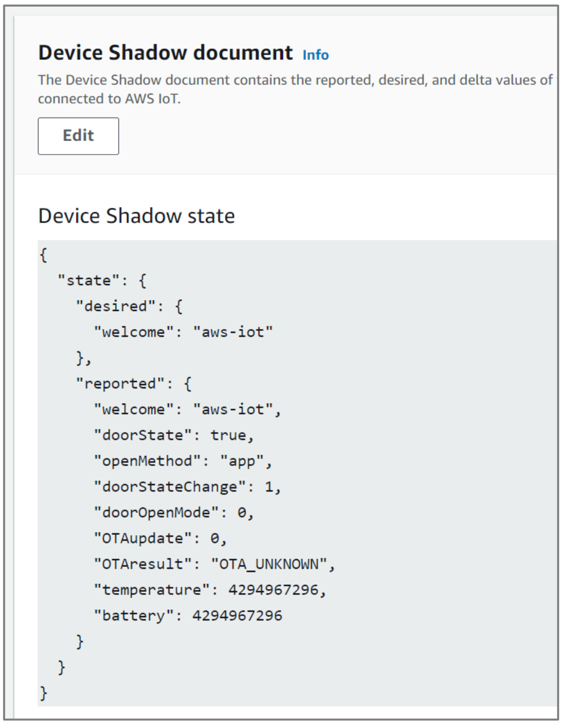
APP-DOORLOCK-1

Battery : - - %
Temperature : - - °C

The door was Opened!

Last reported Time : 2022-10-27 04:03:49 오후

Figure 73. Opened status on application



Device Shadow document

The Device Shadow document contains the reported, desired, and delta values of connected to AWS IoT.

Device Shadow state

```
{
  "state": {
    "desired": {
      "welcome": "aws-iot"
    },
    "reported": {
      "welcome": "aws-iot",
      "doorState": true,
      "openMethod": "app",
      "doorStateChange": 1,
      "doorOpenMode": 0,
      "OTAupdate": 0,
      "OTAresult": "OTA_UNKNOWN",
      "temperature": 4294967296,
      "battery": 4294967296
    }
  }
}
```

Figure 74. Opened status on AWS IoT console

- [Current Status]
  - **Opened**, Battery: _ _%, Temperature: _ _ °C (Real values are displayed on door lock ref. board)
  - Mobile APP (User): **Opened image button**
  - AWS (Server)
    - "doorState": **true**
    - "temperature": 4294967296
    - "battery": 4294967296

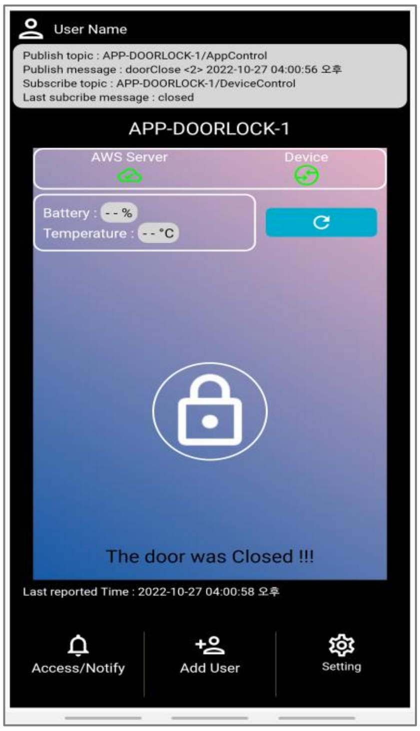
NOTE
A value of 4294967296 for the temperature or battery fields indicates the function is not available.

- DA16200 (Thing): The status of the device is displayed as shown in the **red text**.

```
INFO] [DoorLockDemo] [prvEventCallback:728]

Incoming Publish Topic Name: (Command) APP-DOORLOCK-1/AppControl matches subscribed topic.

Incoming Publish Message : doorOpen


open comm

[openControl]


[INFO] [DoorLockDemo] [controlDoorLock:1555] publish (command response) OK -  payload: "opened"

DEBUG:   [aws_dpm_app_door_work:1974] previous MQTT result = 0, doorLock CMD (=1: 0-idle, 1-open, 2-
close, 3-auto close)
```

```
===========================================================================================


[INFO] [DoorLockDemo] [aws_dpm_app_door_work:2030] publish (shadow doorlock update) OK -        payload:
"{"state":{"reported":{"doorState":true,"openMethod":"app","doorStateChange":1,"doorOpenMode":0,"OTAupdat
e":0,"OTAresult":"OTA_UNKNOWN"}}}"

*********************************************************************************************


last user Timer ID = 5

last doorOpenFlag state: "true"

last FOTA Stat: 0

last FOTA Url: ""
```

7.4.2 Close Door



User Name

Publish topic : APP-DOORLOCK-1/AppControl
Publish message : doorClose <2> 2022-10-27 04:00:56 오후
Subscribe topic : APP-DOORLOCK-1/DeviceControl
Last subcribe message : closed

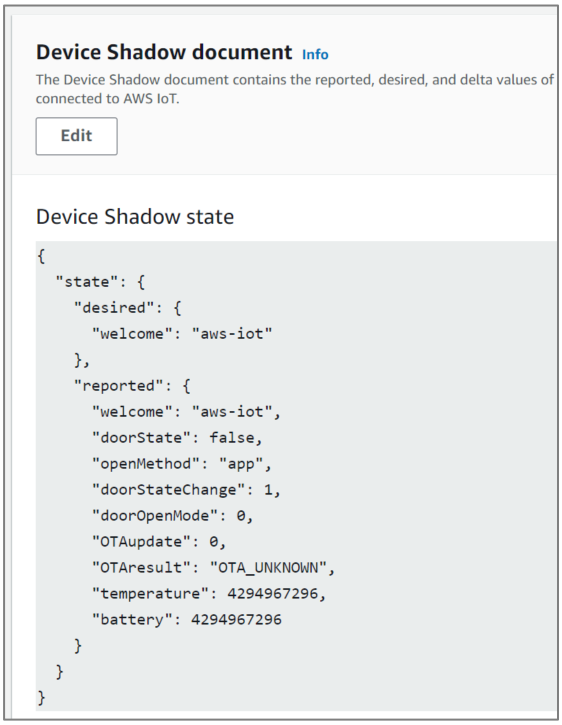
APP-DOORLOCK-1

Battery : - - %
Temperature : - - °C

The door was Closed !!!

Last reported Time : 2022-10-27 04:00:58 오후

Figure 75. Closed status on application



Device Shadow document Info

The Device Shadow document contains the reported, desired, and delta values of connected to AWS IoT.

Device Shadow state

```
{
  "state": {
    "desired": {
      "welcome": "aws-iot"
    },
    "reported": {
      "welcome": "aws-iot",
      "doorState": false,
      "openMethod": "app",
      "doorStateChange": 1,
      "doorOpenMode": 0,
      "OTAupdate": 0,
      "OTAresult": "OTA_UNKNOWN",
      "temperature": 4294967296,
      "battery": 4294967296
    }
  }
}
```

Figure 76. Closed status on AWS IoT console

- [Current Status]
  - **Closed**, Battery: _ _%, Temperature: _ _ °C (Real values are displayed on door lock ref. board)
  - Mobile APP (User): **Closed image button**
  - AWS (Server)
    - "doorState": **false**
    - "temperature": 4294967296
    - "battery": 4294967296

NOTE
A value of 4294967296 for the temperature or battery fields indicates the function is not available.

- DA16200 (Thing): The status of the device is displayed as shown in the **red text**.

```
[INFO] [DoorLockDemo] [prvEventCallback:728]

Incoming Publish Topic Name: (Command) APP-DOORLOCK-1/AppControl matches subscribed topic.

Incoming Publish Message : doorClose
```

```
close comm
[closeControl]

[INFO] [DoorLockDemo] [controlDoorLock:1555] publish (command response) OK -  payload: "closed"
DEBUG:   [aws_dpm_app_door_work:1974] previous MQTT result = 0, doorLock CMD (=2: 0-idle, 1-open, 2-
close, 3-auto close)

=======================================================================================

[INFO] [DoorLockDemo] [aws_dpm_app_door_work:2030] publish (shadow doorlock update) OK -        payload:
"{"state":{"reported":{"doorState":false,"openMethod":"app","doorStateChange":1,"doorOpenMode":0,"OTAupda
te":0,"OTAresult":"OTA_UNKNOWN"}}}"
*****************************************************************************************

last user Timer ID = 5
last doorOpenFlag state: "false"
last FOTA Stat: 0
```

8. OTA Update

Over the Air (OTA) is the process of updating the DA16200/DA16600 firmware image through Wi-Fi using an AWS S3 bucket.

Figure 77 shows the setting up process of the OTA update.

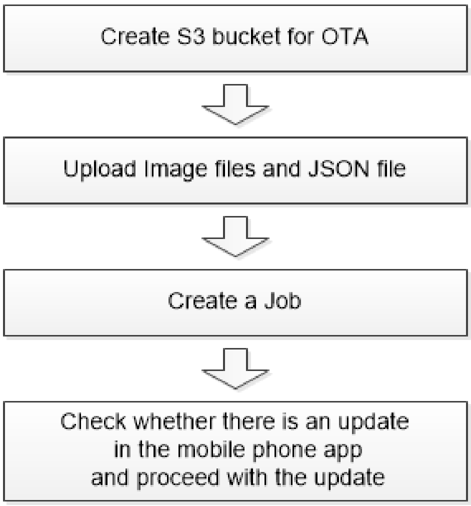
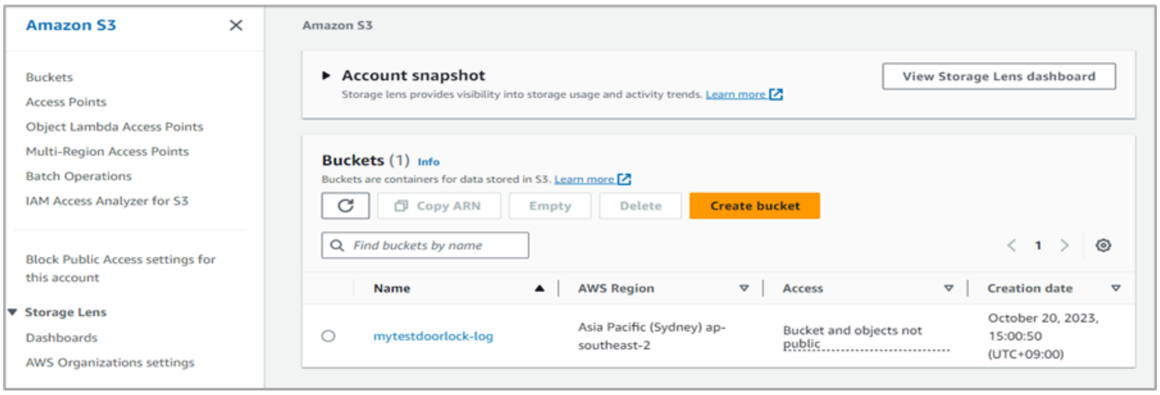

Create S3 bucket for OTA
Upload Image files and JSON file
Create a Job
Check whether there is an update in the mobile phone app and proceed with the update

Figure 77. OTA update

8.1 Create S3 Bucket

For OTA update, create a new bucket in S3:

1. In the Amazon S3 console, click **Create bucket**.


Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Amazon S3
Account snapshot
View Storage Lens dashboard
Buckets (1) Info
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
1

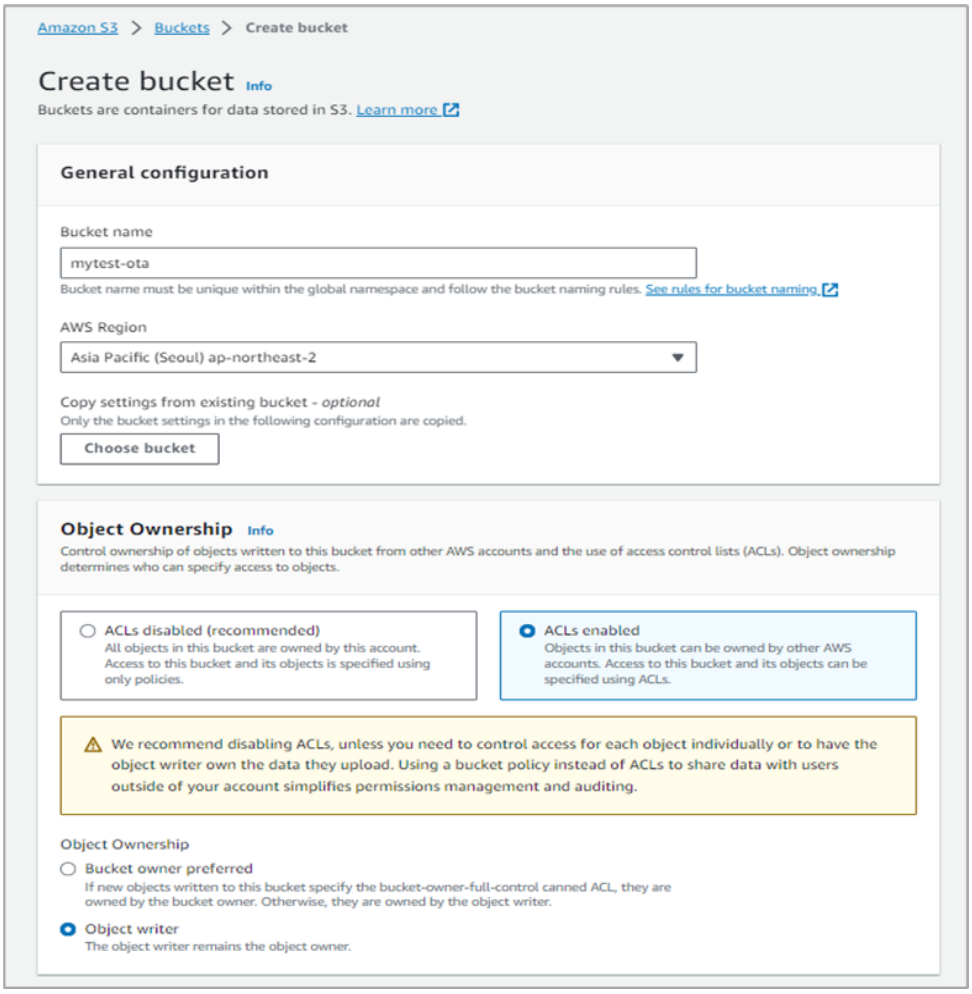
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
○	mytestdoorlock-log	Asia Pacific (Sydney) ap-southeast-2	Bucket and objects not public	October 20, 2023, 15:00:50 (UTC+09:00)

Figure 78. Create bucket for OTA update

2. Enter a Bucket name, apply the settings as shown in Figure 79–Figure 81, and click **Create bucket**.



Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name: mytest-ota

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

AWS Region: Asia Pacific (Seoul) ap-northeast-2

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

○ ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

● ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

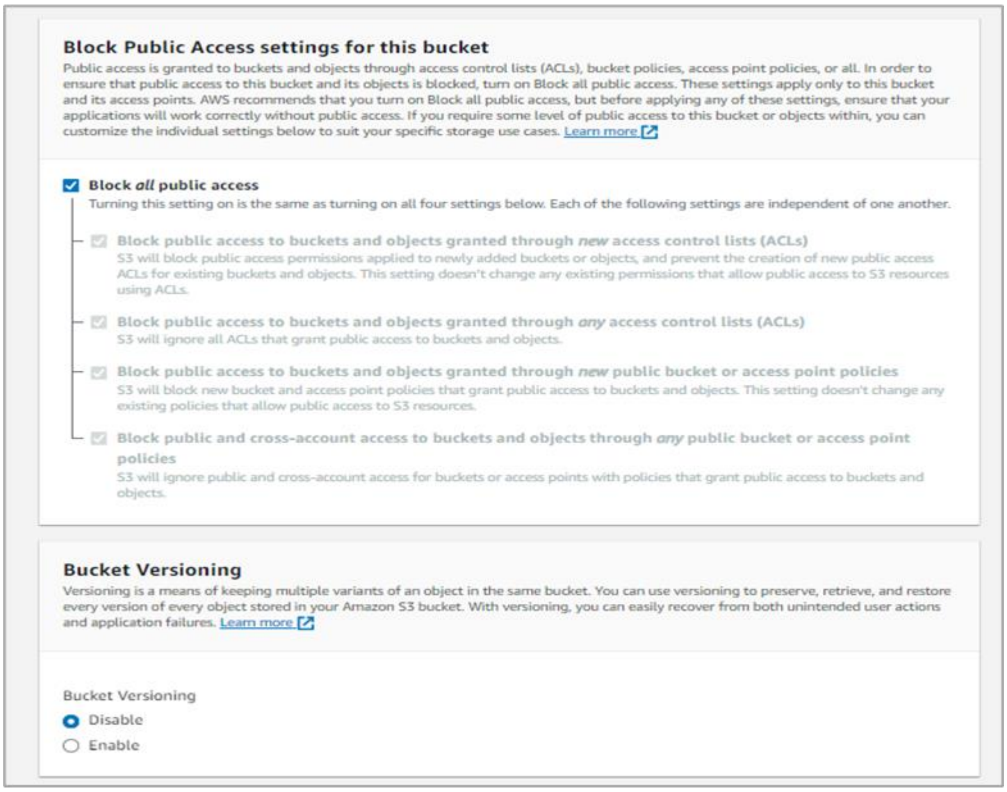
⚠ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

Object Ownership

○ Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

● Object writer
The object writer remains the object owner.

Figure 79. Bucket configuration – general and object ownership



Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

☑ **Block *all* public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

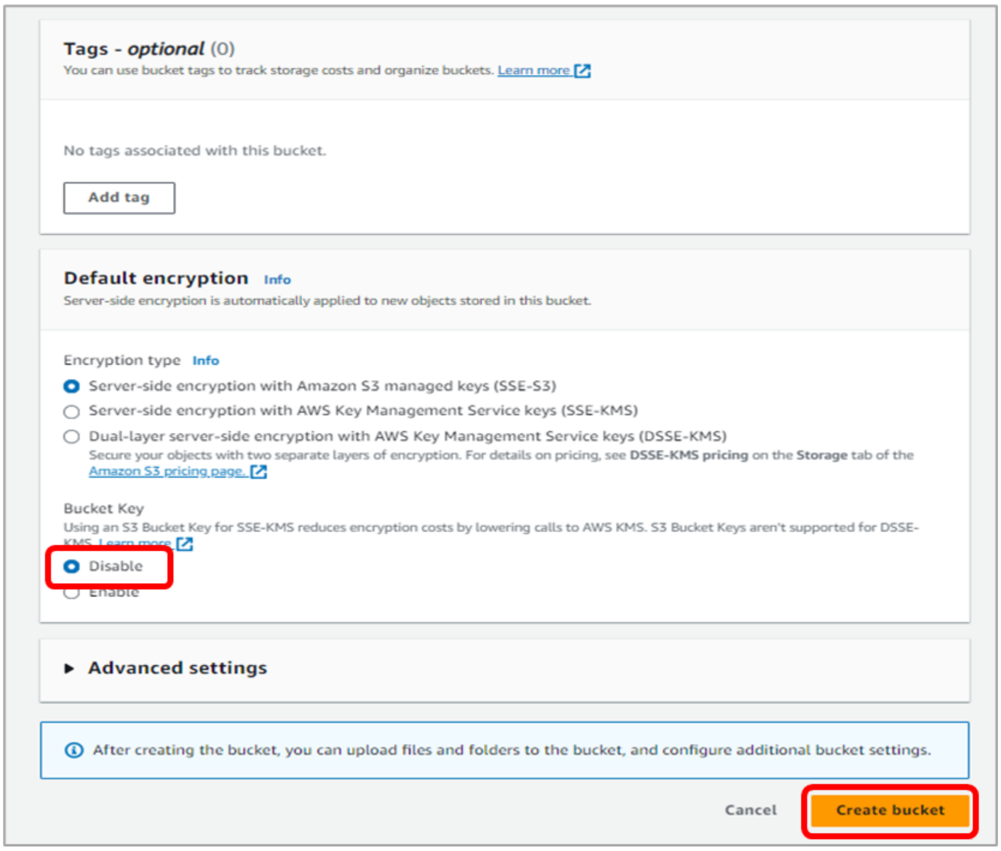
- ☑ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
  S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ☑ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
  S3 will ignore all ACLs that grant public access to buckets and objects.
- ☑ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
  S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ☑ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
  S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning
● Disable
○ Enable

Figure 80. Bucket configuration – public access and versioning



Tags - *optional* (0)

You can use bucket tags to track storage costs and organize buckets. Learn more

No tags associated with this bucket.

Default encryption Info

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type Info
● Server-side encryption with Amazon S3 managed keys (SSE-S3)
○ Server-side encryption with AWS Key Management Service keys (SSE-KMS)
○ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see **DSSE-KMS pricing** on the **Storage** tab of the Amazon S3 pricing page.

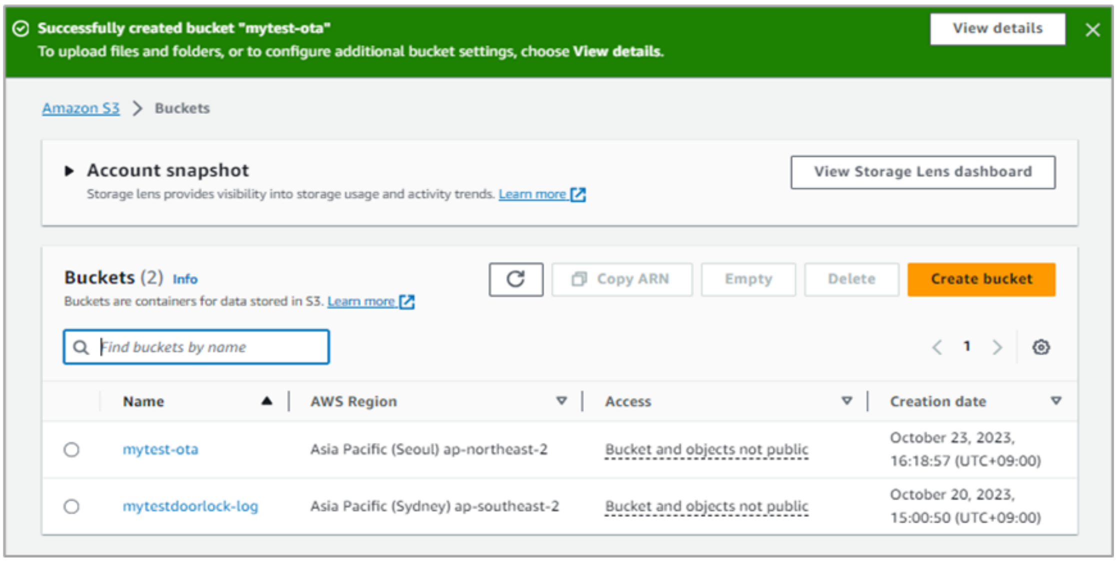
Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more
● Disable
○ Enable

▸ **Advanced settings**

ⓘ After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Figure 81. Bucket configuration – bucket key

3. Select the created bucket in the Buckets list.



Successfully created bucket "mytest-ota"
To upload files and folders, or to configure additional bucket settings, choose **View details**.

Amazon S3 > Buckets

▸ Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (2) Info
Buckets are containers for data stored in S3. Learn more

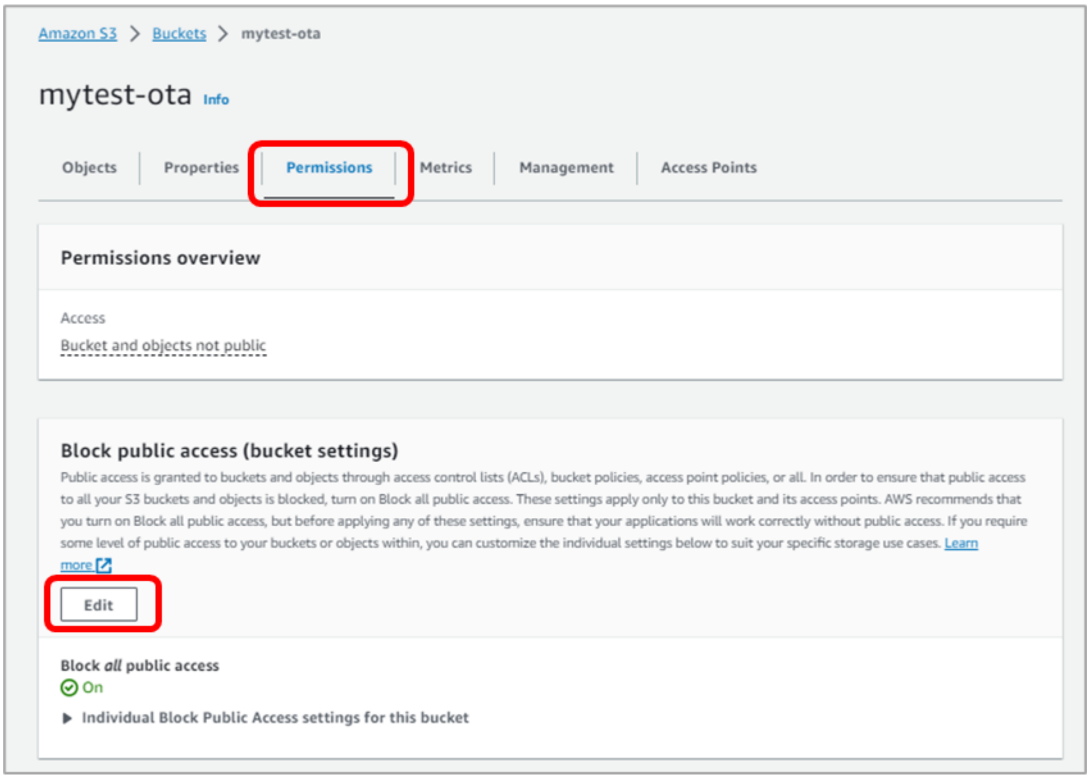
	Name ▲	AWS Region	Access	Creation date
○	mytest-ota	Asia Pacific (Seoul) ap-northeast-2	Bucket and objects not public	October 23, 2023, 16:18:57 (UTC+09:00)
○	mytestdoorlock-log	Asia Pacific (Sydney) ap-southeast-2	Bucket and objects not public	October 20, 2023, 15:00:50 (UTC+09:00)

Figure 82. Created buckets for OTA

4. Click the **Permissions** tab, and then click **Edit**.

   This bucket must be modified for public access in the next step.

NOTE
Use public buckets for development environments only due to security concerns.



Amazon S3 > Buckets > mytest-ota

mytest-ota Info

Permissions overview

Access
Bucket and objects not public

Block public access (bucket settings)

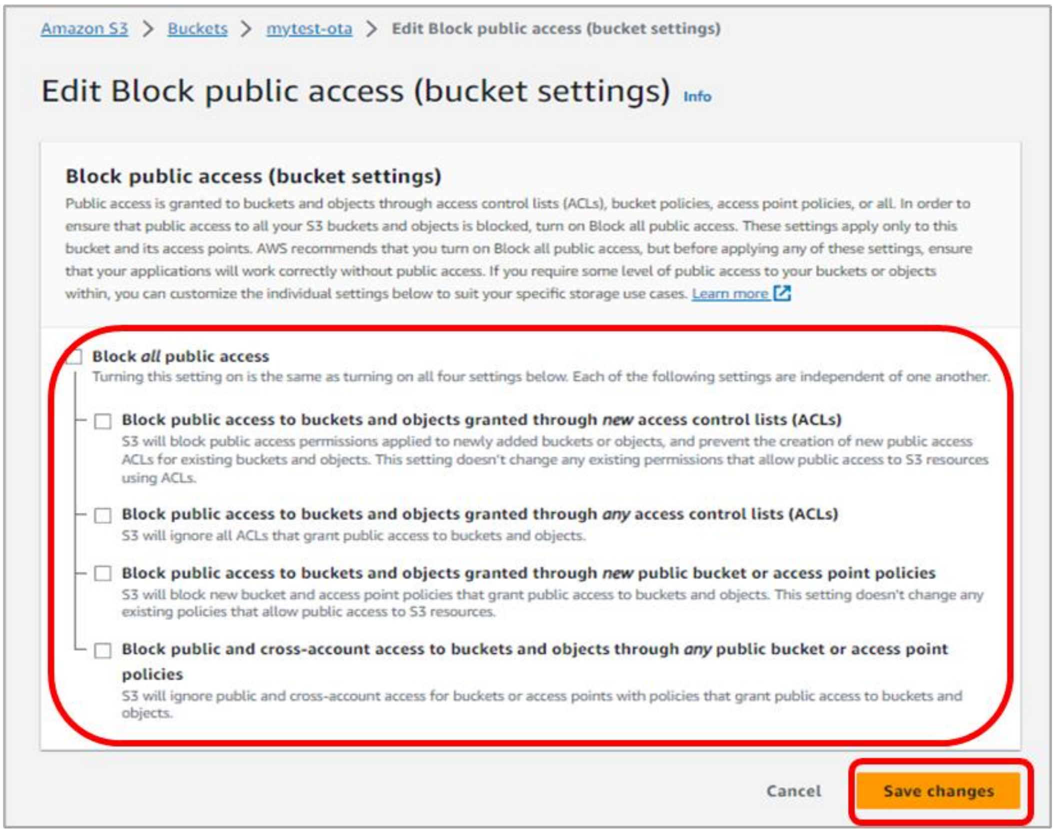
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

Block *all* public access
✓ On

▸ Individual Block Public Access settings for this bucket

Figure 83. Edit bucket for public access

5. Clear all checkboxes, and then click **Save changes**.



Amazon S3 > Buckets > mytest-ota > Edit Block public access (bucket settings)

Edit Block public access (bucket settings) Info

Block public access (bucket settings)

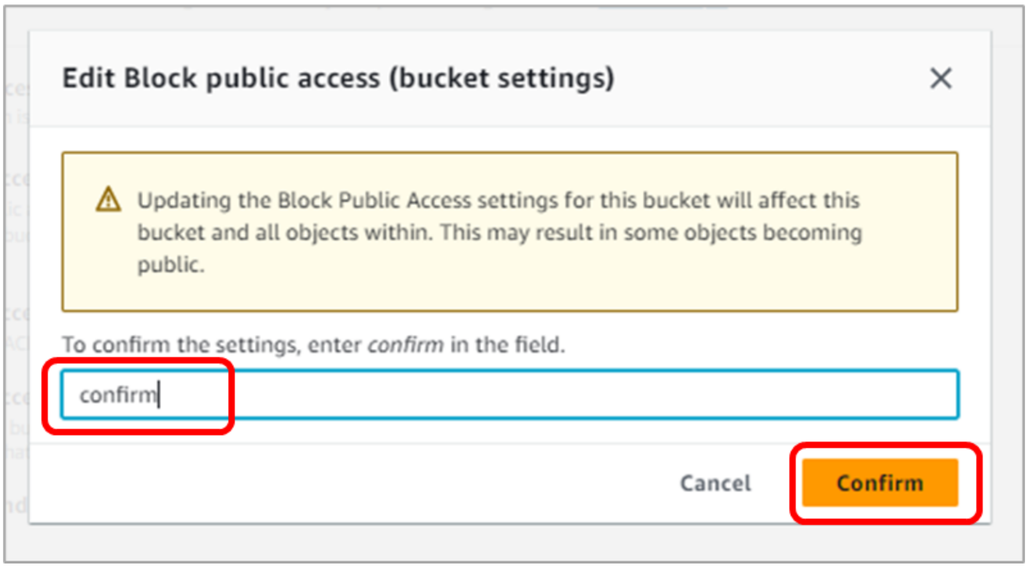
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

□ Block *all* public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- □ Block public access to buckets and objects granted through *new* access control lists (ACLs)
  S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- □ Block public access to buckets and objects granted through *any* access control lists (ACLs)
  S3 will ignore all ACLs that grant public access to buckets and objects.
- □ Block public access to buckets and objects granted through *new* public bucket or access point policies
  S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- □ Block public and cross-account access to buckets and objects through *any* public bucket or access point policies
  S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Figure 84. Public access settings for bucket

6. To save the settings, type “confirm”, and then click **Confirm**.



Edit Block public access (bucket settings)

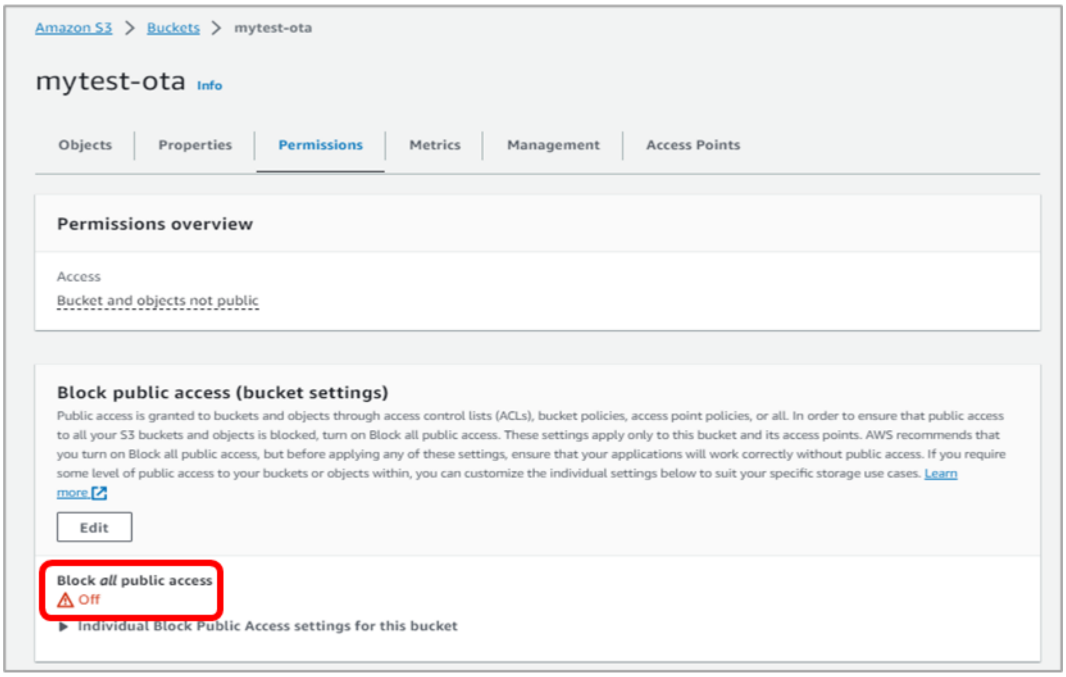
Updating the Block Public Access settings for this bucket will affect this bucket and all objects within. This may result in some objects becoming public.

To confirm the settings, enter *confirm* in the field.

confirm

Figure 85. Confirm settings

7. On the **Permissions** tab, verify that all block options of public access are off.



Permissions overview

Access

Bucket and objects not public

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

Block *all* public access

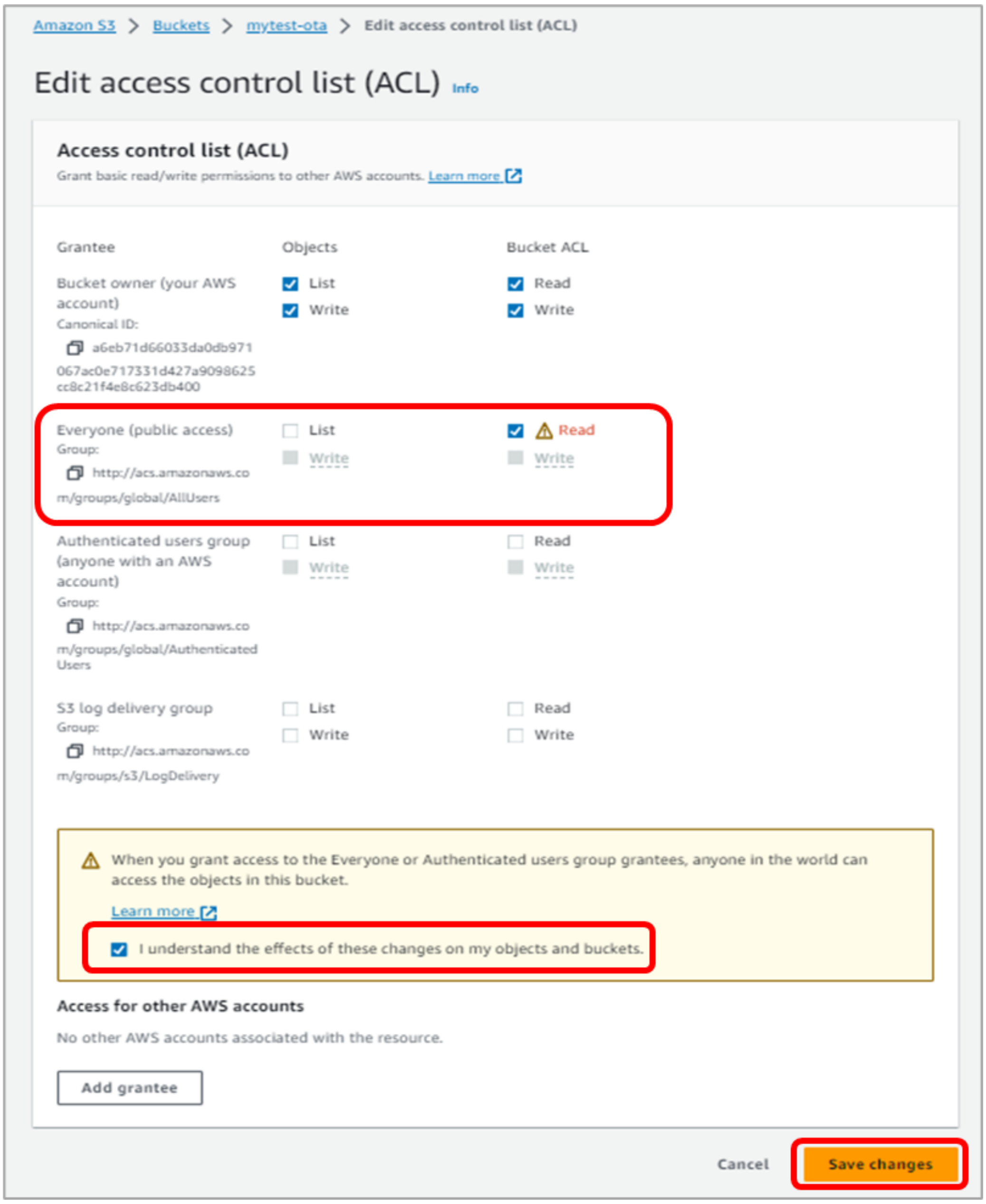
⚠ Off

▶ Individual Block Public Access settings for this bucket

Figure 86. Settings updated

8. Click **Access Control List (ACL)** > **Edit** and next to **Everyone**, select **Read** Bucket ACL, and then click **Save changes**.

NOTE
To avoid ACLs, visit https://docs.aws.amazon.com/AmazonS3/latest/userguide/about-object-ownership.html.



Amazon S3 > Buckets > mytest-ota > Edit access control list (ACL)

Edit access control list (ACL) Info

Access control list (ACL)

Grant basic read/write permissions to other AWS accounts. Learn more

Grantee	Objects	Bucket ACL
Bucket owner (your AWS account) Canonical ID: a6eb71d66033da0db971067ac0e717331d427a9098625cc8c21f4e8c623db400	☑ List ☑ Write	☑ Read ☑ Write
Everyone (public access) Group: http://acs.amazonaws.com/groups/global/AllUsers	□ List □ Write	☑ ⚠ Read □ Write
Authenticated users group (anyone with an AWS account) Group: http://acs.amazonaws.com/groups/global/AuthenticatedUsers	□ List □ Write	□ Read □ Write
S3 log delivery group Group: http://acs.amazonaws.com/groups/s3/LogDelivery	□ List □ Write	□ Read □ Write

⚠ When you grant access to the Everyone or Authenticated users group grantees, anyone in the world can access the objects in this bucket.

Learn more

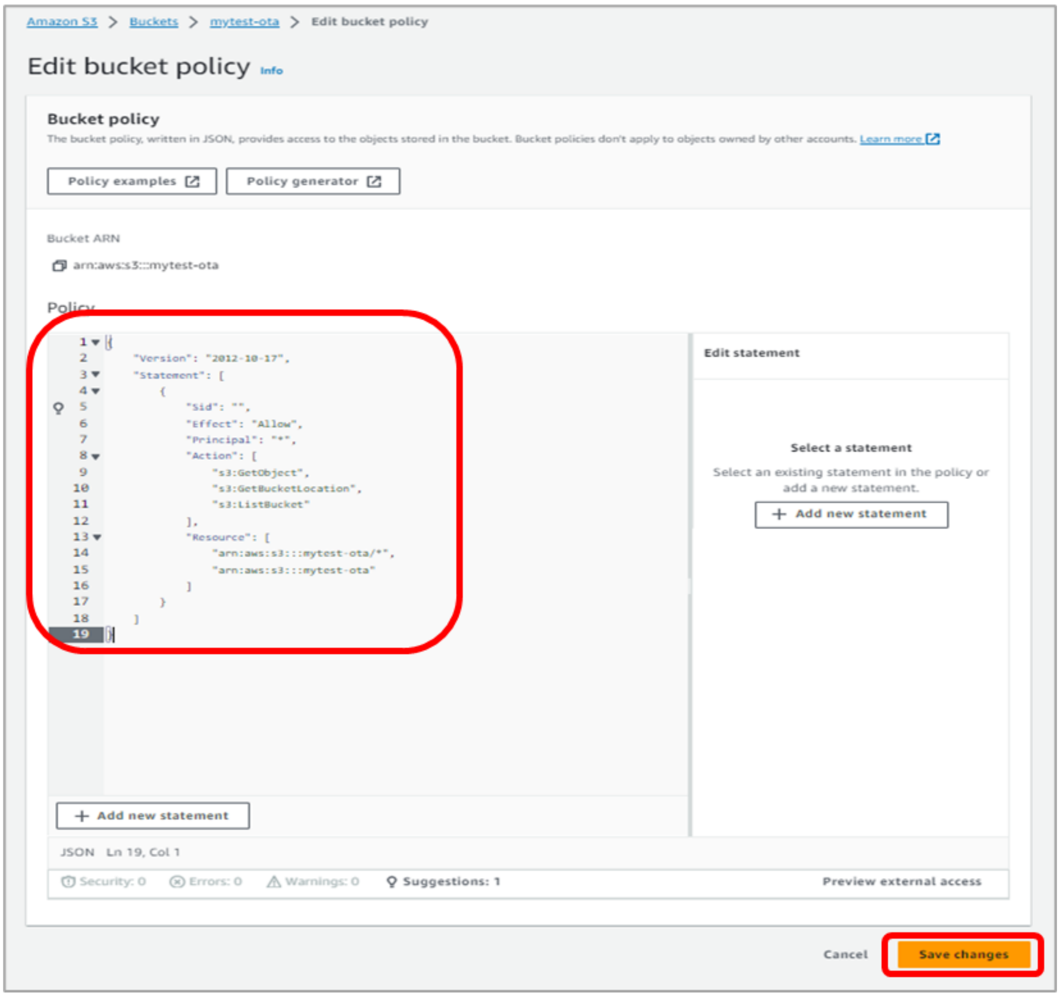
☑ I understand the effects of these changes on my objects and buckets.

Access for other AWS accounts

No other AWS accounts associated with the resource.

Figure 87. Public access for everyone

9. The bucket policy must be added as shown in Figure 88 and Table 5.
   "User Bucket Name" in Table 5 is the name of the S3 bucket created for an OTA update.



Amazon S3 > Buckets > mytest-ota > Edit bucket policy

Edit bucket policy Info

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

Bucket ARN

arn:aws:s3:::mytest-ota

Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::mytest-ota/*",
                "arn:aws:s3:::mytest-ota"
            ]
        }
    ]
}
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

JSON Ln 19, Col 1

Security: 0 Errors: 0 Warnings: 0 Suggestions: 1 Preview external access

Figure 88. Bucket policy editor

Table 5. Bucket policy in JSON format

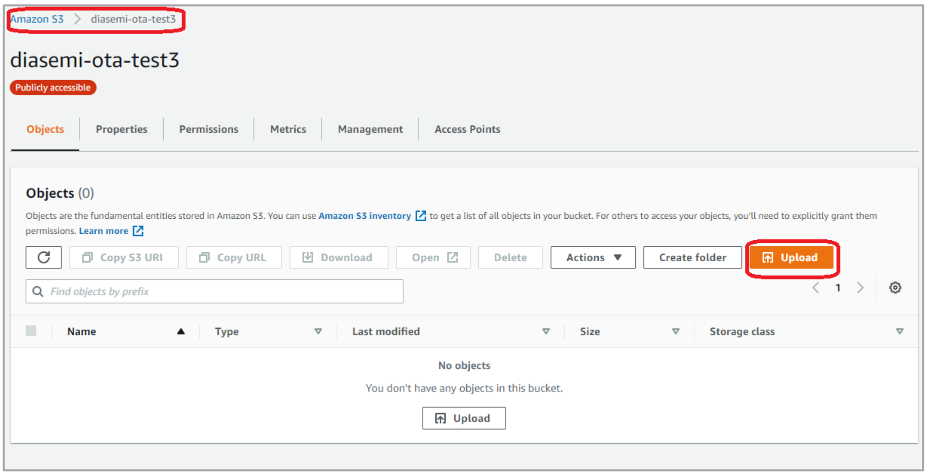
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::User Bucket Name/*",
                "arn:aws:s3:::User Bucket Name"
            ]
        }
    ]
}
```

8.2 Upload Image File and JSON File

1. Rename the image files as follows:

- RTOS Image: DA16200_FRTOS-GEN01.img

2. To upload the image files and JSON file for the OTA update, click **Upload**.



Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

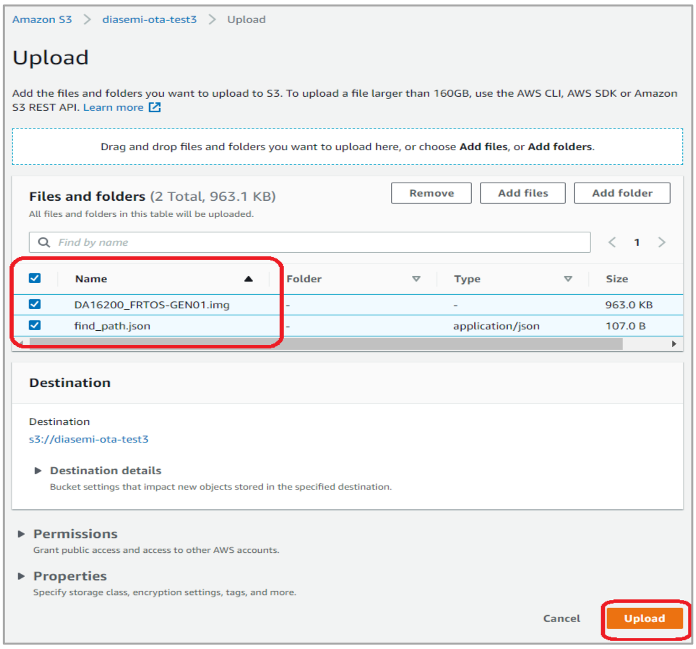
☐	Name ▲	Type	Last modified	Size	Storage class

No objects

You don't have any objects in this bucket.

Figure 89. Upload files

3. Drag and drop or add files to upload.
4. There is one IMG file for DA16200 OTA update, and the JSON file is a path setting file for the update. The important thing is that the names of the two files for the update should be the same as in Figure 90.



Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folders**.

Files and folders (2 Total, 963.1 KB)

All files and folders in this table will be uploaded.

☑	Name ▲	Folder	Type	Size
☑	DA16200_FRTOS-GEN01.img	-	-	963.0 KB
☑	find_path.json	-	application/json	107.0 B

Destination

Destination
s3://diasemi-ota-test3

▸ Destination details
Bucket settings that impact new objects stored in the specified destination.

▸ Permissions
Grant public access and access to other AWS accounts.

▸ Properties
Specify storage class, encryption settings, tags, and more.

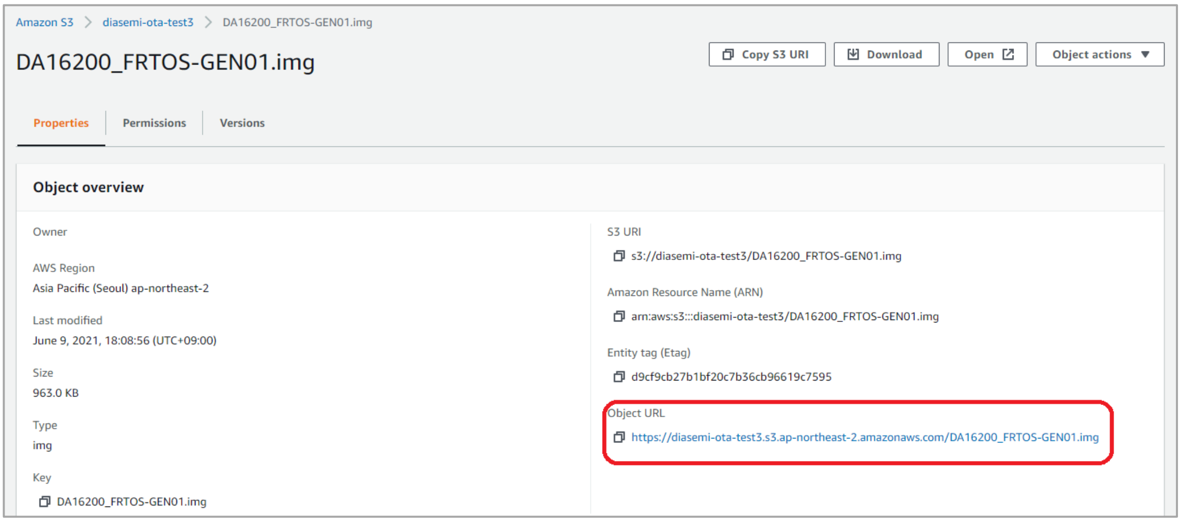
Figure 90. Ready to upload

The JSON information for the OTA update is as follows:

```
{ "operation":"install",

"Source":"https:// User Bucket Name.s3.ap-northeast-2.amazonaws.com/" }
```

"User Bucket Name" is the name of the S3 bucket created for the OTA update. The URL policy of the "Source" can be changed by AWS.

5. Click the uploaded file name to check it.



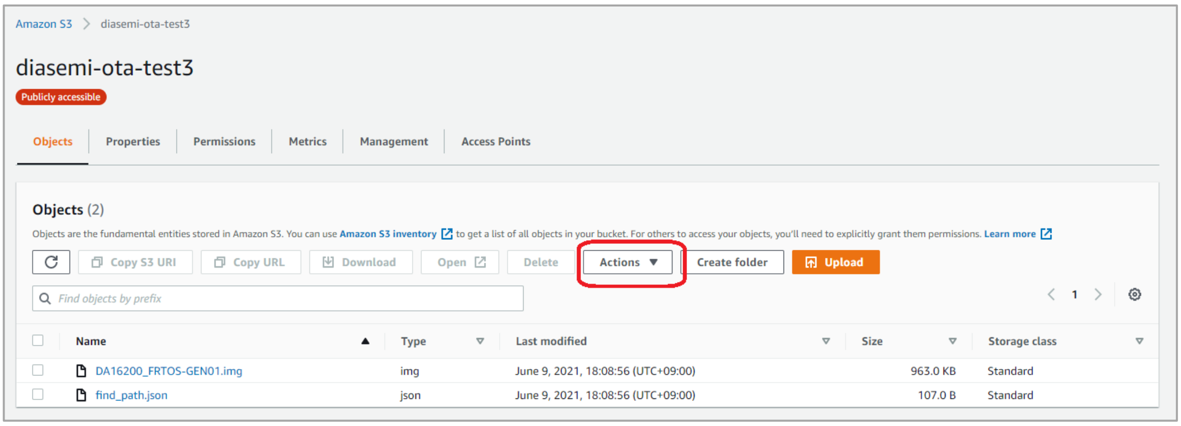
DA16200_FRTOS-GEN01.img

Object overview

Owner	S3 URI s3://diasemi-ota-test3/DA16200_FRTOS-GEN01.img
AWS Region Asia Pacific (Seoul) ap-northeast-2	Amazon Resource Name (ARN) arn:aws:s3:::diasemi-ota-test3/DA16200_FRTOS-GEN01.img
Last modified June 9, 2021, 18:08:56 (UTC+09:00)	Entity tag (Etag) d9cf9cb27b1bf20c7b36cb96619c7595
Size 963.0 KB	Object URL https://diasemi-ota-test3.s3.ap-northeast-2.amazonaws.com/DA16200_FRTOS-GEN01.img
Type img	
Key DA16200_FRTOS-GEN01.img	

Figure 91. URL of source

6. Check if the files are uploaded correctly. You can delete and/or reupload files to the bucket on the **Actions** tab.



diasemi-ota-test3

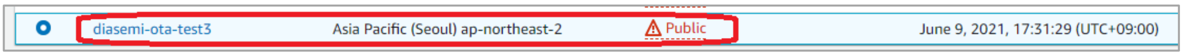
Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	DA16200_FRTOS-GEN01.img	img	June 9, 2021, 18:08:56 (UTC+09:00)	963.0 KB	Standard
□	find_path.json	json	June 9, 2021, 18:08:56 (UTC+09:00)	107.0 B	Standard

Figure 92. Uploaded files

As a result, a publicly accessible bucket is created.



● diasemi-ota-test3 Asia Pacific (Seoul) ap-northeast-2 ⚠ Public June 9, 2021, 17:31:29 (UTC+09:00)

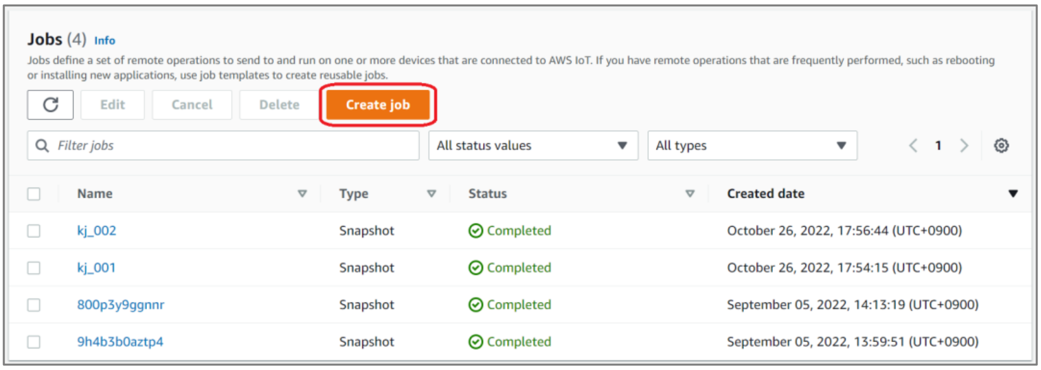
Figure 93. Completed setup for OTA update

8.3 Create Job

AWS IoT Jobs is a service that allows you to define a set of remote operations that are sent to and executed on one or more devices connected to AWS IoT.

For an OTA update, go to the **IoT Core** service in AWS Management Console. OTA is the process of replacing a product with a newer version of the same product. A Job must be created and registered to perform an OTA update. It is a task to access the file uploaded to the bucket of the S3 service. If the server operator registers this Job at the desired time, the test thing proceeds with the OTA update.

1. In the AWS Management Console, go to **IoT core** > **Manage > Remote Actions** > **Jobs,** and click **Create job**.Figure 94. Create job



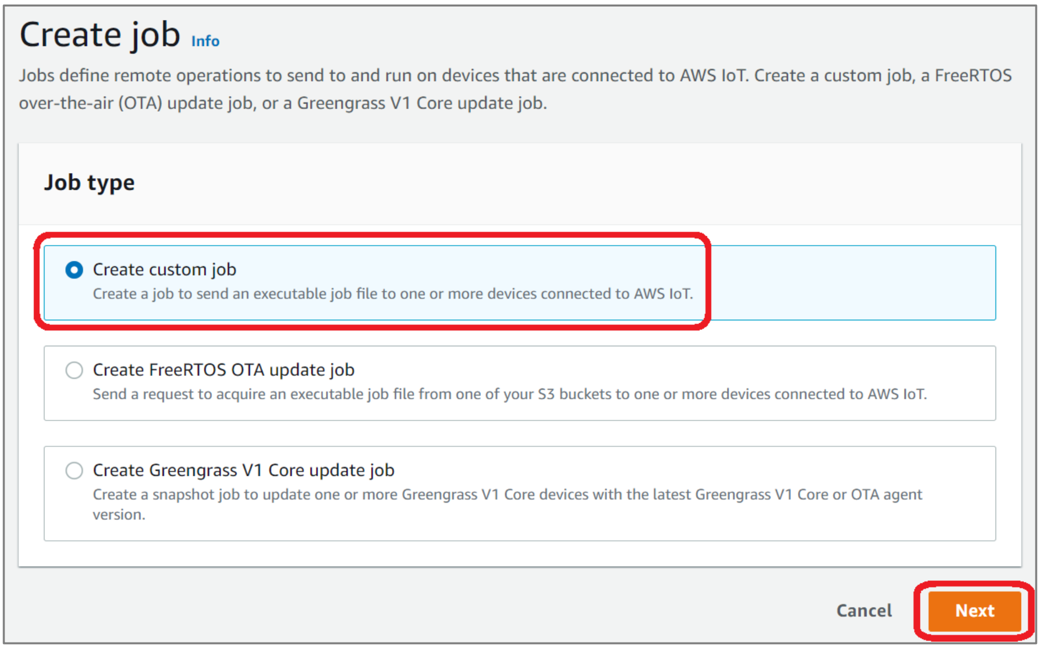
Jobs (4) Info

Jobs define a set of remote operations to send to and run on one or more devices that are connected to AWS IoT. If you have remote operations that are frequently performed, such as rebooting or installing new applications, use job templates to create reusable jobs.

□	Name	Type	Status	Created date
□	kj_002	Snapshot	✓ Completed	October 26, 2022, 17:56:44 (UTC+0900)
□	kj_001	Snapshot	✓ Completed	October 26, 2022, 17:54:15 (UTC+0900)
□	800p3y9ggnnr	Snapshot	✓ Completed	September 05, 2022, 14:13:19 (UTC+0900)
□	9h4b3b0aztp4	Snapshot	✓ Completed	September 05, 2022, 13:59:51 (UTC+0900)

Figure 94. Create job

2. Select **Create custom job** and click **Next**.



Create job Info

Jobs define remote operations to send to and run on devices that are connected to AWS IoT. Create a custom job, a FreeRTOS over-the-air (OTA) update job, or a Greengrass V1 Core update job.

Job type

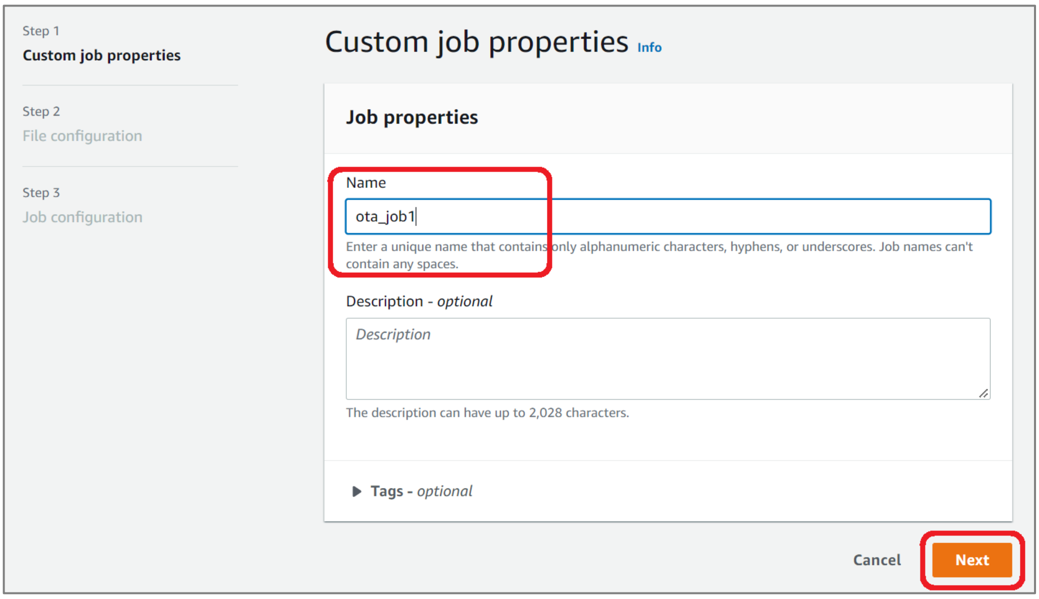
● Create custom job
Create a job to send an executable job file to one or more devices connected to AWS IoT.

○ Create FreeRTOS OTA update job
Send a request to acquire an executable job file from one of your S3 buckets to one or more devices connected to AWS IoT.

○ Create Greengrass V1 Core update job
Create a snapshot job to update one or more Greengrass V1 Core devices with the latest Greengrass V1 Core or OTA agent version.

Figure 95. Create custom job

3. In the **Name** field, enter the job name and click **Next**.



Step 1
Custom job properties

Step 2
File configuration

Step 3
Job configuration

Custom job properties Info

Job properties

Name: ota_job1

Enter a unique name that contains only alphanumeric characters, hyphens, or underscores. Job names can't contain any spaces.

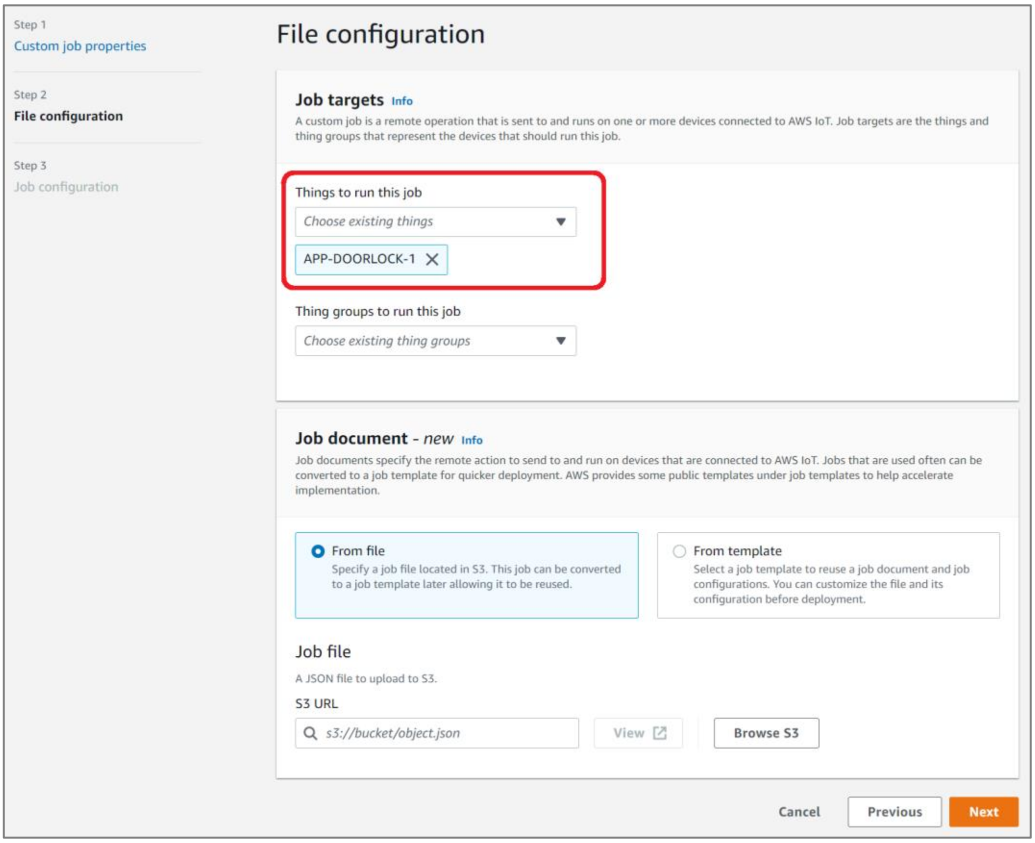
Description - *optional*

The description can have up to 2,028 characters.

▶ Tags - *optional*

Figure 96. Enter job name

4. Select the devices to update. The thing to select is available in the list of options.



Step 1
Custom job properties

Step 2
File configuration

Step 3
Job configuration

File configuration

Job targets Info

A custom job is a remote operation that is sent to and runs on one or more devices connected to AWS IoT. Job targets are the things and thing groups that represent the devices that should run this job.

Things to run this job: APP-DOORLOCK-1

Thing groups to run this job

Job document - *new* Info

Job documents specify the remote action to send to and run on devices that are connected to AWS IoT. Jobs that are used often can be converted to a job template for quicker deployment. AWS provides some public templates under job templates to help accelerate implementation.

● From file
Specify a job file located in S3. This job can be converted to a job template later allowing it to be reused.

○ From template
Select a job template to reuse a job document and job configurations. You can customize the file and its configuration before deployment.

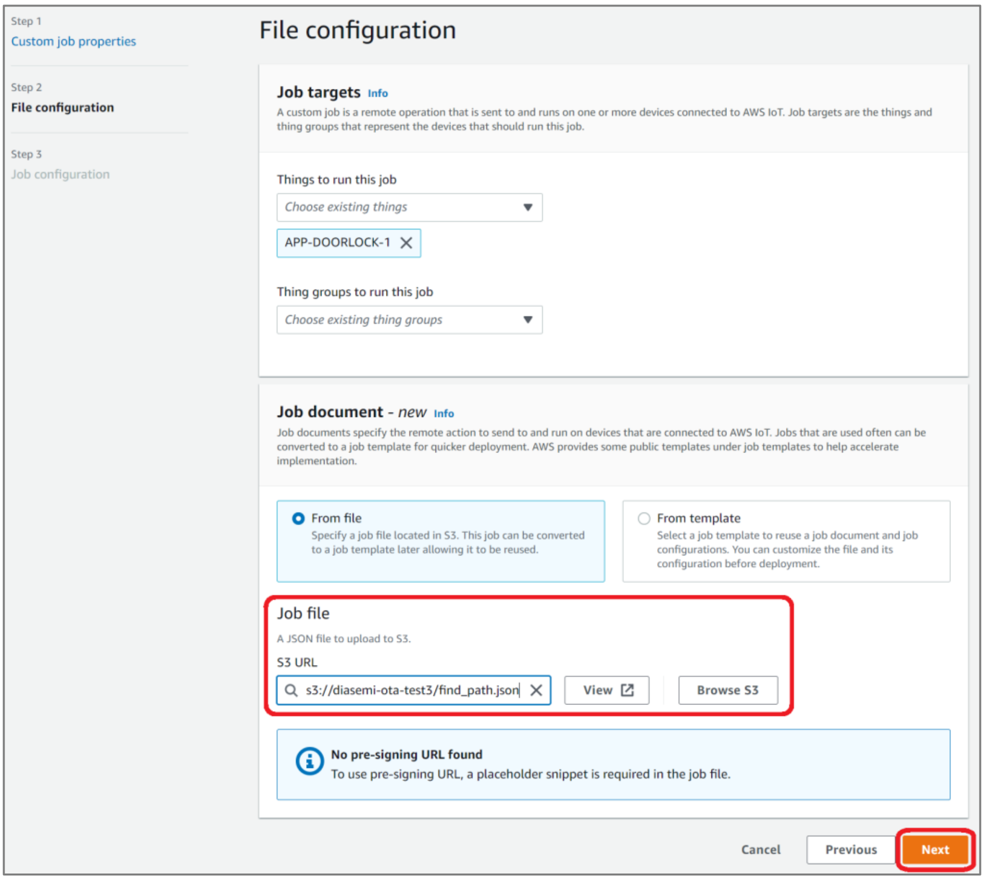
Job file

A JSON file to upload to S3.

S3 URL

Figure 97. Select thing for OTA update

5. Under **Job file**, click **Browse S3** and select the S3 URL and click **Next**.



Step 1
Custom job properties

Step 2
File configuration

Step 3
Job configuration

File configuration

Job targets Info

A custom job is a remote operation that is sent to and runs on one or more devices connected to AWS IoT. Job targets are the things and thing groups that represent the devices that should run this job.

Things to run this job

Choose existing things ▼

APP-DOORLOCK-1 ✕

Thing groups to run this job

Choose existing thing groups ▼

Job document - *new* Info

Job documents specify the remote action to send to and run on devices that are connected to AWS IoT. Jobs that are used often can be converted to a job template for quicker deployment. AWS provides some public templates under job templates to help accelerate implementation.

● From file
Specify a job file located in S3. This job can be converted to a job template later allowing it to be reused.

○ From template
Select a job template to reuse a job document and job configurations. You can customize the file and its configuration before deployment.

Job file

A JSON file to upload to S3.

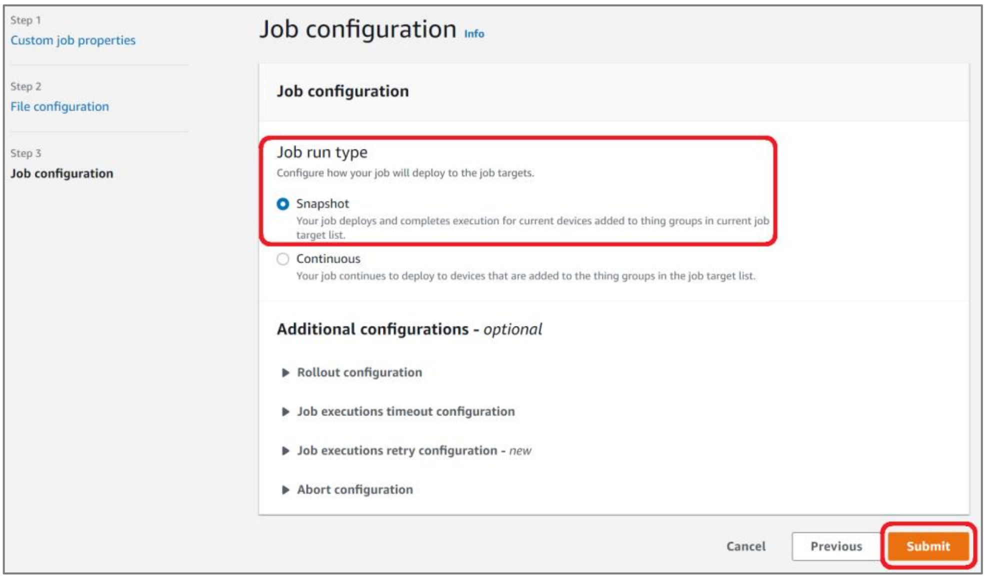
S3 URL

s3://diasemi-ota-test3/find_path.json ✕ | View | Browse S3

No pre-signing URL found
To use pre-signing URL, a placeholder snippet is required in the job file.

Figure 98. Select JSON for OTA update

6. Under **Job run type**, select **Snapshot** and click **Submit**.



Step 1
Custom job properties

Step 2
File configuration

Step 3
Job configuration

Job configuration Info

Job configuration

Job run type

Configure how your job will deploy to the job targets.

● Snapshot
Your job deploys and completes execution for current devices added to thing groups in current job target list.

○ Continuous
Your job continues to deploy to devices that are added to the thing groups in the job target list.

Additional configurations - *optional*

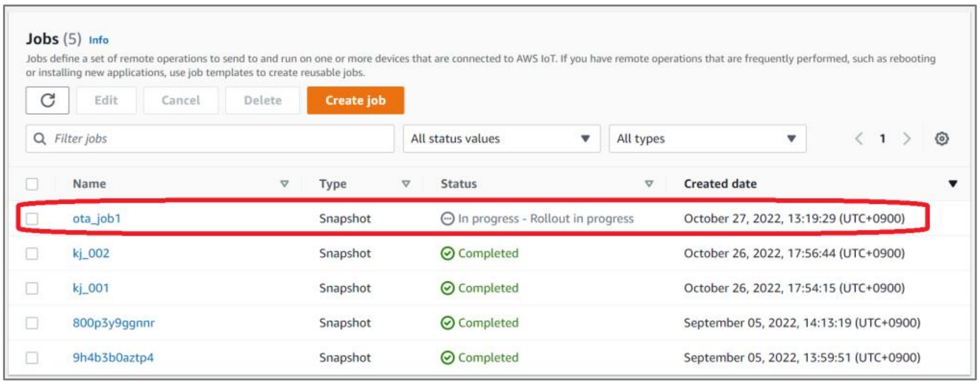
▶ Rollout configuration

▶ Job executions timeout configuration

▶ Job executions retry configuration - *new*

▶ Abort configuration

Figure 99. Job run type

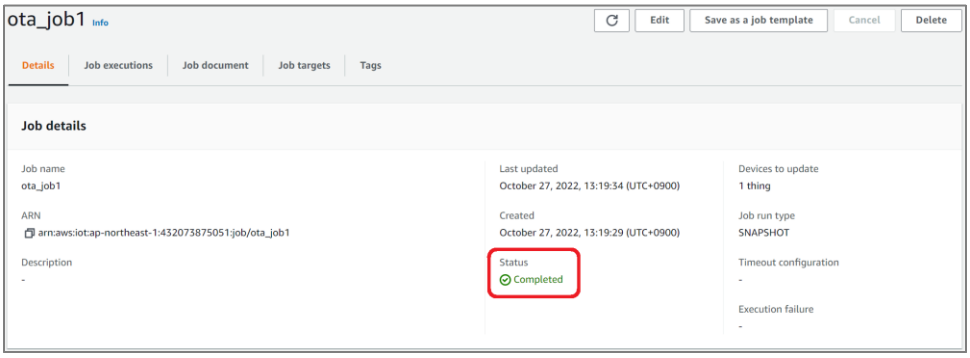


Jobs (5) Info

Jobs define a set of remote operations to send to and run on one or more devices that are connected to AWS IoT. If you have remote operations that are frequently performed, such as rebooting or installing new applications, use job templates to create reusable jobs.

□	Name	Type	Status	Created date
□	ota_job1	Snapshot	In progress - Rollout in progress	October 27, 2022, 13:19:29 (UTC+0900)
□	kj_002	Snapshot	✓ Completed	October 26, 2022, 17:56:44 (UTC+0900)
□	kj_001	Snapshot	✓ Completed	October 26, 2022, 17:54:15 (UTC+0900)
□	800p3y9ggnnr	Snapshot	✓ Completed	September 05, 2022, 14:13:19 (UTC+0900)
□	9h4b3b0aztp4	Snapshot	✓ Completed	September 05, 2022, 13:59:51 (UTC+0900)

Figure 100. Job being created



ota_job1 Info

Job details

Job name
ota_job1

ARN
arn:aws:iot:ap-northeast-1:432073875051:job/ota_job1

Description
-

Last updated
October 27, 2022, 13:19:34 (UTC+0900)

Created
October 27, 2022, 13:19:29 (UTC+0900)

Status
✓ Completed

Devices to update
1 thing

Job run type
SNAPSHOT

Timeout configuration
-

Execution failure
-

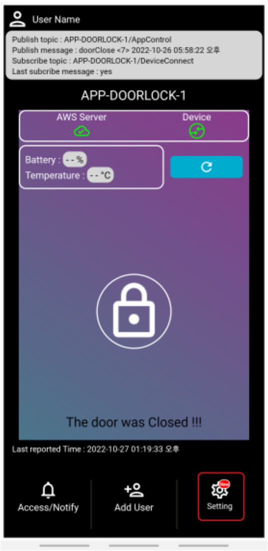
Figure 101. Successfully created job

8.4 Execute OTA Update

When a job is created successfully, the device receives the job details as follows:

```
[dpmAPPManager] DM_NEED_CONNECTION
DM_NEED_CONNECTION


[INFO] [DoorLockDemo] [aws_dpm_app_connect:2267] Establishing MQTT session with provisioned
certificate...
recv timeout(=2000 ms) set OK (socket=0)
hostName = "a1kzdt4nun8bnh-ats.iot.ap-northeast-1.amazonaws.com", flag to re-query (=0)
host IP from RTM = "54.178.218.11"
TCP connection OK to "a1kzdt4nun8bnh-ats.iot.ap-northeast-1.amazonaws.com"
[INFO] [DoorLockDemo] [aws_dpm_app_connect:2317] Sucessfully established connection with provisioned
credentials.
[Make AWS-Thing-Name]
[NVRAM] AWS Thing name : [APP-DOORLOCK-1] (len=14)
[NVRAM] [APP-DOORLOCK-1/DeviceConnect][APP-DOORLOCK-1/AppControl][APP-DOORLOCK-1/DeviceControl]
[INFO] [DoorLockDemo] [aws_dpm_app_subscription:1939] subscription info: total(default:4, tried:4), OK(4)
current RTM user Timer ID = 5
current RTM temperature(str): 0.000000
current RTM battery(str): 0.000000
current RTM doorOpen state: "false"
current RTM doorOpenMode : 0
current RTM FOTAFlag: 1
current RTM FOTA url : "https://diasemi-ota-test3.s3.ap-northeast-2.amazonaws.com/"
[dpmAPPManager] DM_RTC_WAKEUP
DM_WAKEUP_TIMER (tid=5)


DEBUG:   [aws_dpm_app_sensor_work:2104] read values from sensor if available


=======================================================================================
recv timeout(=120 ms) set OK (socket=0)
[INFO] [DoorLockDemo] [aws_dpm_app_sensor_work:2162] publish (shadow sensor update) OK -        payload:
"{"state":{"reported":{"doorState":false,"temperature":4294967296.000000,"battery":4294967296.000000}}}"
***************************************************************************************


last temperature: Not available
last battery: Not available
Sleep mode 3: KA timer interval(=1800 sec)
DM_FINISH_DEVICE


recv timeout(=20 ms) set OK (socket=0)
[dpm_keepalive_timer_register] RTC interval (=1780 secs), mode (=0)
>>> Start DPM Power-Down !!!
```

NOTE
▪ When a Job for an OTA update is created, you can see the URL of the S3 bucket accessed through JSON in the console. Also, the setting icon changes in the Mobile application. See Figure 102 and the console message. ▪ The temperature and battery value displayed as 4294967296 indicates that it is not available.


User Name

Publish topic : APP-DOORLOCK-1/AppControl
Publish message : doorClose <7> 2022-10-26 05:58:22 오후
Subscribe topic : APP-DOORLOCK-1/DeviceConnect
Last subcribe message : yes

APP-DOORLOCK-1

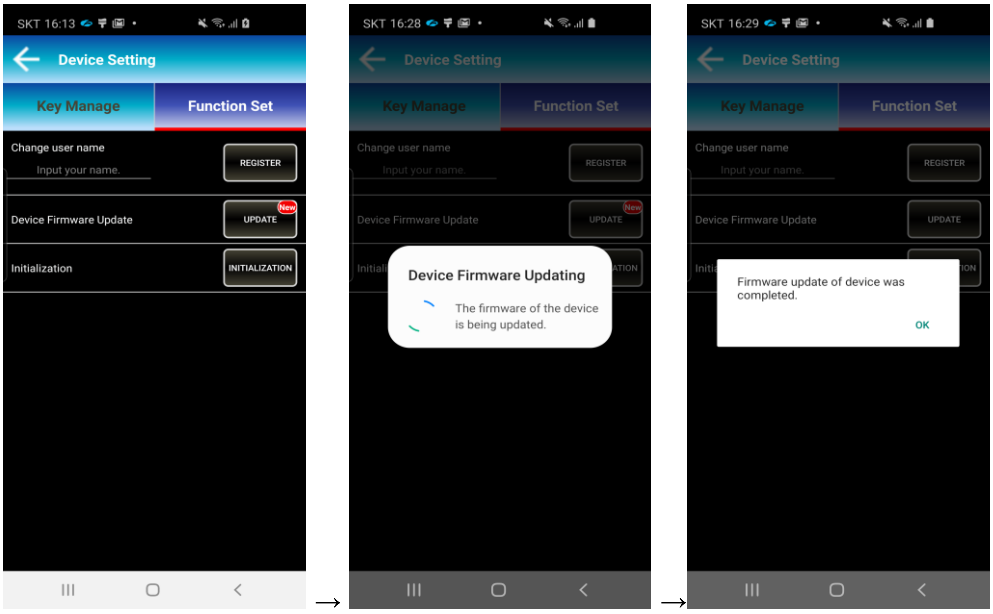
Battery : -- %
Temperature : -- °C

The door was Closed !!!

Last reported Time : 2022-10-27 01:19:33 오후

Figure 102. Successful job for OTA update in mobile app

The update is executed when you click the **Update** button on the Setting screen. The console and the Android application show the progress status during the OTA update. When the update is completed, the thing restarts, and in the Android device, the update notification disappears (Figure 103).


Device Firmware Updating

The firmware of the device is being updated.

Firmware update of device was completed.

OK

Figure 103. Execute OTA update in Android app

The following example shows the console message when an update is being performed.

```
last user Timer ID = 5

last doorOpenFlag state: "false"

last FOTA Stat: 2

last FOTA Url: "https://diasemi-ota-test3.s3.ap-northeast-2.amazonaws.com/"

URL for updating: "https://diasemi-ota-test3.s3.ap-northeast-2.amazonaws.com/"
```

```
save URL info & reboot for OTA

Wakeup source is 0x0

...

...

...

DEBUG:   [aws_ota_fw_update:3532]  RTOS url https://diasemi-ota-test2.s3.ap-northeast-
2.amazonaws.com/DA16200_FRTOS-GEN01.img

>>> SNTP Server: pool.ntp.org (106.247.248.106)

>>> SNTP Time sync : 2022.10.26 - 08:56:58

> Server FW version : FRTOS-GEN01-01-56c232799-004457

   >> HTTP(s) Client Downloading... 100%(1202848/1202848 Bytes)

- OTA Update : <RTOS> Download - Success

DEBUG:   [app_ota_fw_download_complete_notify:3375] RTOS download finish. (0x00)

- OTA: Renewing with new F/W

- OTA: RTOS

        > Same Version : FRTOS-GEN01-01-56c232799-004457

>>> RTOS is updated and system reboots. (New boot_idx=0) !!!

DEBUG:   [app_ota_fw_renew_notify:3497] Succeeded to replace with new FW.

- OTA: Reboot after 0 secs ...

Wakeup source is 0x0

[dpm_init_retmemory] DPM INIT CONFIGURATION(1)
```

9. Private S3 Download Demo

This section explains how to download a private S3 bucket object content using DA16200. For the demonstration, AWS Signature Version 4 (Authenticating Requests (AWS Signature Version 4) - Amazon Simple Storage Service) is used for authentication.

9.1 Sign Up for AWS Account

For the Demo, you can create a new AWS account by following steps in Section 6.1.1 or use the AWS account created for this demonstration.

Use the following credential for signing in.

```
Username: wifiapps0@gmail.com
Password: Wifiapp@123
```

9.2 Create S3 Bucket

For private S3 bucket file download demo, you need to create a S3 bucket.

1. In the S3 console, select **Create bucket**.

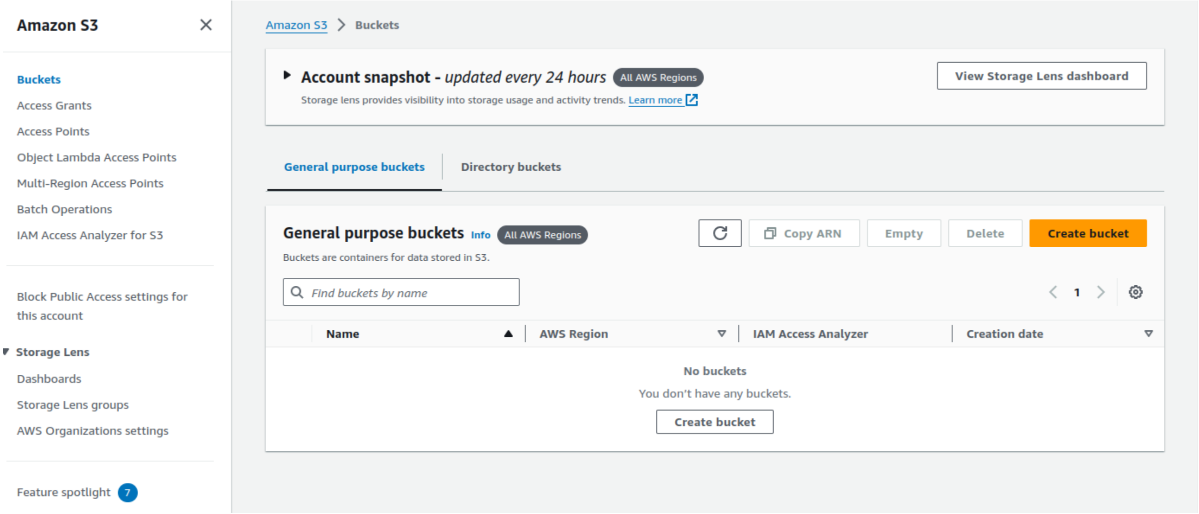
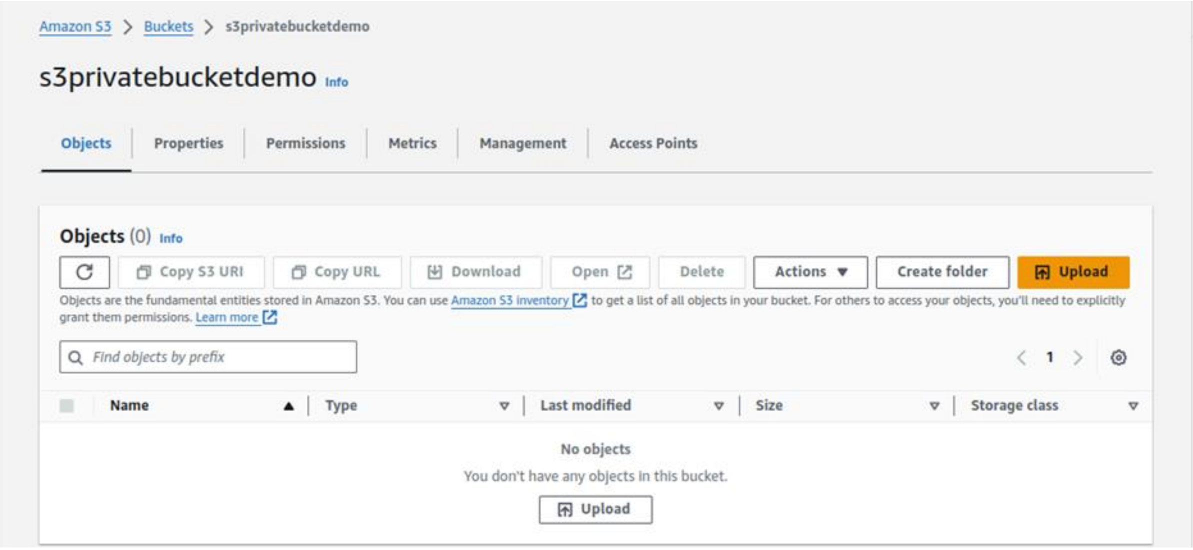

Amazon S3
×
Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 7
Amazon S3 > Buckets
Account snapshot - updated every 24 hours All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
General purpose buckets
Directory buckets
General purpose buckets Info All AWS Regions
Buckets are containers for data stored in S3.
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
1
Name
AWS Region
IAM Access Analyzer
Creation date
No buckets
You don't have any buckets.
Create bucket

Figure 104. Create private bucket

2. Enter a Bucket name and select **Create bucket**.
3. In the S3 bucket, you need to upload the files you want to download. Choose **Upload** after selecting the bucket.

NOTE
While creating buckets, all S3 buckets are private by default.



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

No objects

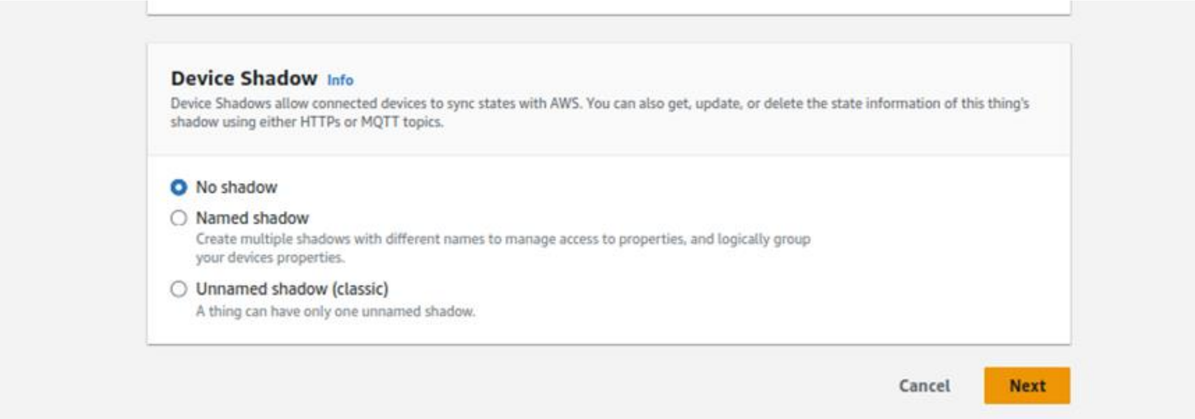
You don't have any objects in this bucket.

Figure 105. Upload files to bucket

4. Drag and drop or add files to upload.
5. Select **Upload**.

9.3 Create AWS IoT Thing and Certificate

1. To create a thing, follow the steps 1 to 5 in Section 6.1.2.1 and select **No shadow**.



Device Shadows allow connected devices to sync states with AWS. You can also get, update, or delete the state information of this thing's shadow using either HTTPs or MQTT topics.

● No shadow

○ Named shadow
Create multiple shadows with different names to manage access to properties, and logically group your devices properties.

○ Unnamed shadow (classic)
A thing can have only one unnamed shadow.

Figure 106. Device shadow in thing

2. To create and configure certificate to thing, follow steps in Sections 6.1.2.2, 6.1.2.3, and 6.1.2.4.

NOTE
Names for thing and policy can be modified to intended use.

9.4 Create Policies

You can manage access in AWS by creating policies and attaching them to IAM identities.

To create policy:

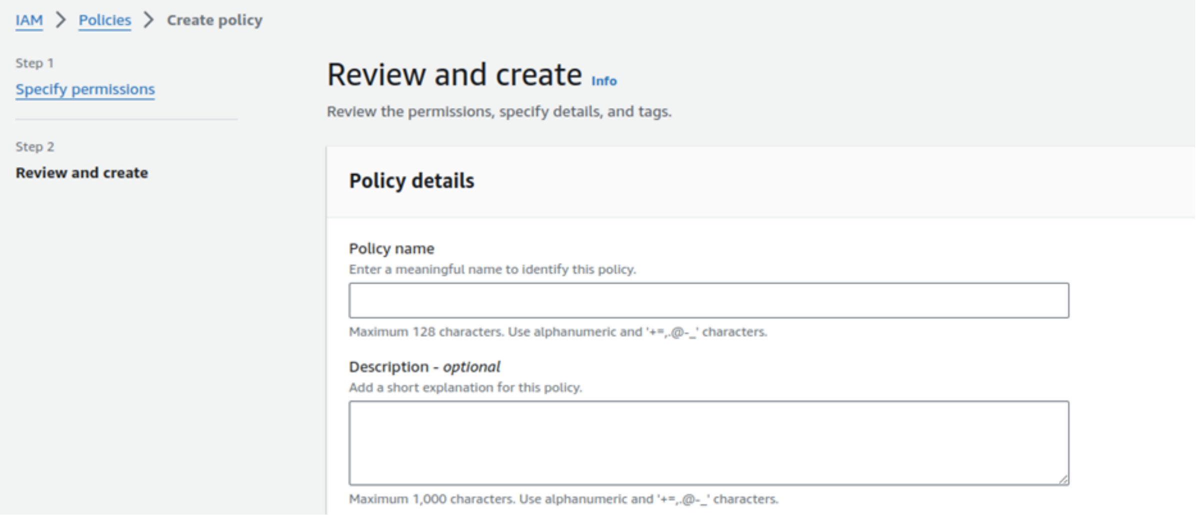
1. Go to the **IAM console** in dropdown and select **Policies**.
2. Select **Create policy**.
3. Copy the following JSON which is used for performing actions on S3.

```
{
    "Version": "2012-10-17",
    "Statement": {
```

```
    "Effect": "Allow",
    "Action": [
            "s3:GetObject"
            ],
    "Resource": "arn:aws:s3:::<BUCKET_NAME>/*"
     }
 }
```

NOTE
<BUCKET_NAME>: Use the bucket name created previously.

4. Select **Next**.
5. Give policy a name.



IAM > Policies > Create policy

Step 1
Specify permissions

Step 2
Review and create

Review and create Info

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.

Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Description - *optional*
Add a short explanation for this policy.

Maximum 1,000 characters. Use alphanumeric and '+=,.@-_' characters.

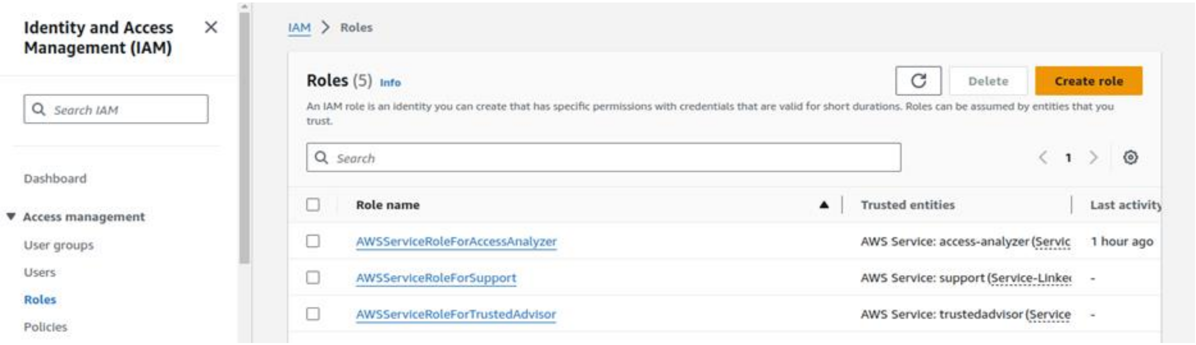
Figure 107. Create private policy

6. Select **Create**.

9.5 Create and Configure IAM Role

An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role and an IAM user are similar, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS.

1. Go to the **IAM console** and select **Roles**.



Roles (5) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

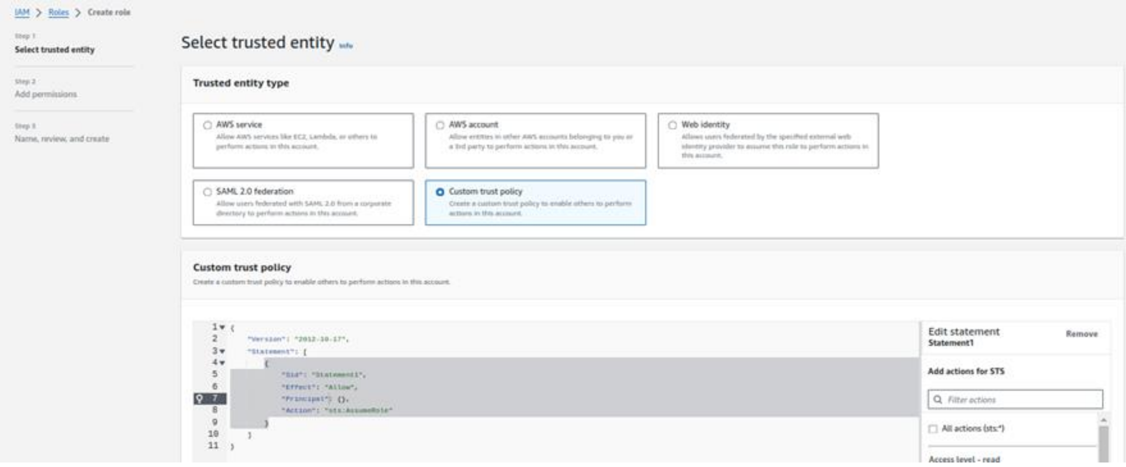
□	Role name	Trusted entities	Last activity
□	AWSServiceRoleForAccessAnalyzer	AWS Service: access-analyzer (Servic	1 hour ago
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-

Figure 108. Create private role

2. Select **Create role**.
3. Select custom trust policy and copy the following trust policy (JSON) for grants the credentials provider permission to assume the role.

```
{
    "Version": "2012-10-17",
    "Statement": {
    "Effect": "Allow",
    "Principal": {"Service": "credentials.iot.amazonaws.com"},
    "Action": "sts:AssumeRole"
     }
 }
```

4. Select **Next**



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity Info

Trusted entity type

○ AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

● Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": {},
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Edit statement
Statement1

Add actions for STS

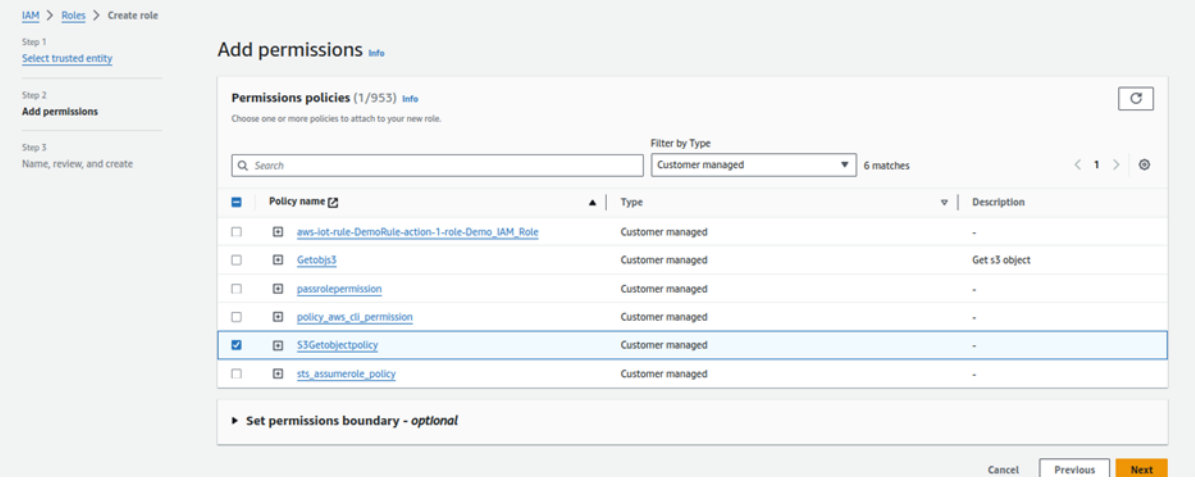
Filter actions

□ All actions (sts:*)

Access level - read

Figure 109. Trusted entity

5. In the **Filter by type** drop down, choose **Customer managed** and select the previously created policy in Section 9.4.



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions Info

Permissions policies (1/953) Info
Choose one or more policies to attach to your new role.

Search

Filter by Type: Customer managed ▼ 6 matches

		Policy name	Type	Description
□	⊞	aws-iot-rule-DemoRule-action-1-role-Demo_IAM_Role	Customer managed	-
□	⊞	Getobjs3	Customer managed	Get s3 object
□	⊞	passrolepermission	Customer managed	-
□	⊞	policy_aws_cli_permission	Customer managed	-
☑	⊞	S3Getobjectpolicy	Customer managed	-
□	⊞	sts_assumerole_policy	Customer managed	-

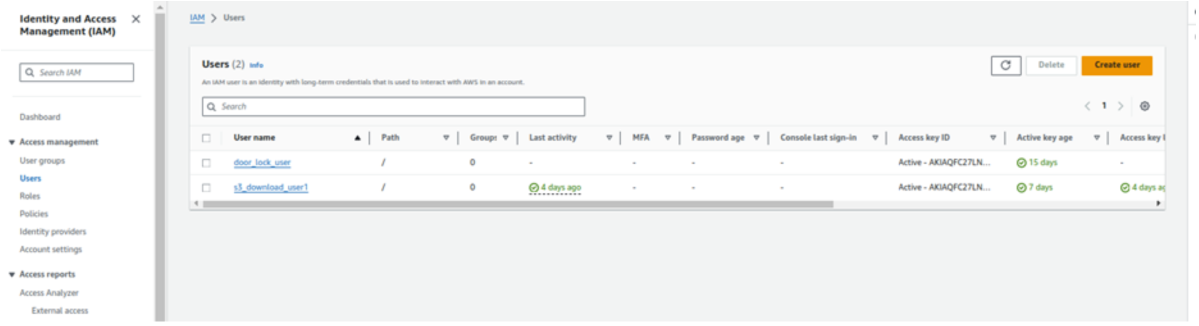
▸ **Set permissions boundary - *optional***

Figure 110. Adding policy to role

6. Also, select **AWSIotFullAccess** policy in **All type** filter.
7. Select **Next.**
8. Enter a name for the role and select Create role.

9.6 Create and Configure IAM User

1. In the IAM console, select **Users** > **Create User**.



Identity and Access Management (IAM)

Search IAM

Dashboard

▾ Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

▾ Access reports

Access Analyzer

External access

IAM > Users

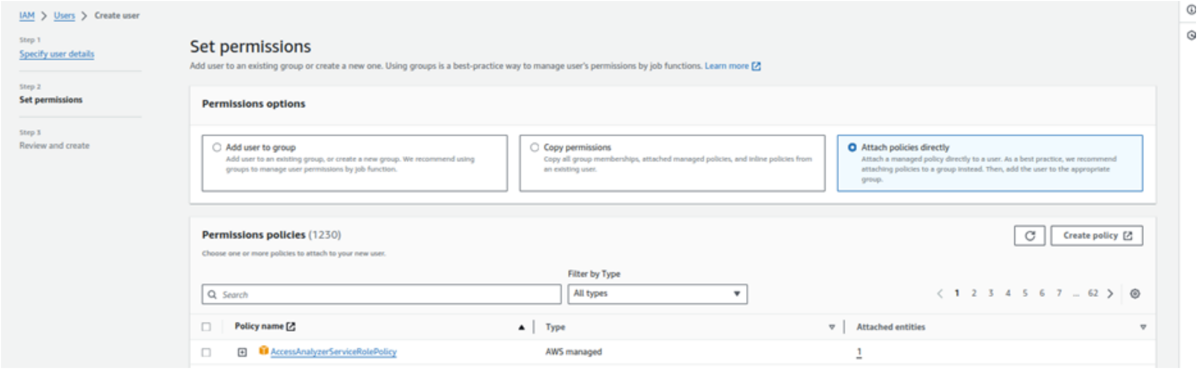
Users (2) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

□	User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Access key ID	Active key age	Access key l
□	door_lock_user	/	0	-	-	-	-	Active - AKIAQFC27LN...	✓ 15 days	-
□	s3_download_user1	/	0	✓ 4 days ago	-	-	-	Active - AKIAQFC27LN...	✓ 7 days	✓ 4 days ag

Figure 111. Create private user

2. Enter a **User name and select Next.**
3. Select Attach policies directly.



IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Permissions options

○ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

○ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

● Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1230)

Choose one or more policies to attach to your new user.

□	Policy name	Type	Attached entities
□	AccessAnalyzerServiceRolePolicy	AWS managed	1

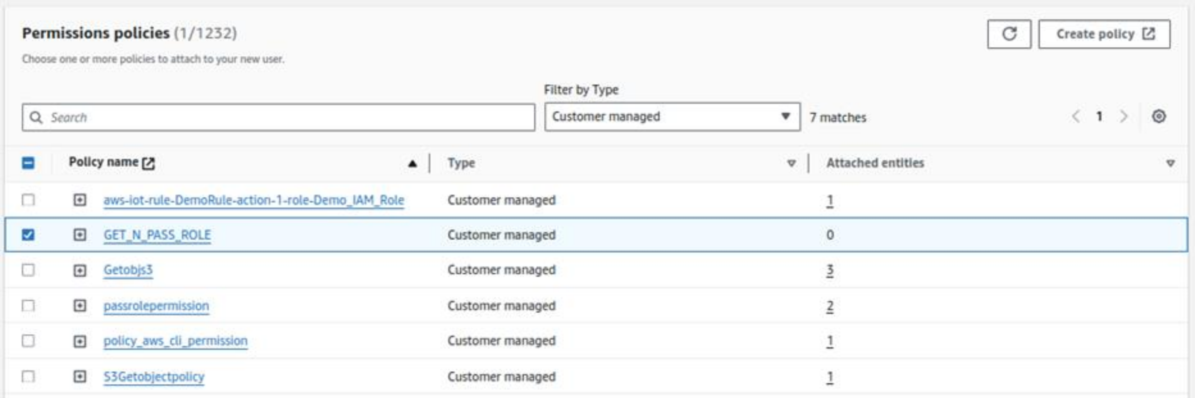
Figure 112. Permission setting

4. Select **Create policy**.
5. The IAM role that you have created must be passed to AWS IoT to create a role alias. For that, follow Section 9.4 with the given JSON.

```
{
            "Version": "2012-10-17",
            "Statement": {
              "Effect": "Allow",
              "Action": [
                  "iam:GetRole",
                  "iam:PassRole"
              ],
              "Resource": "arn:aws:iam::<your_aws_account_id>:role/<IAM_ROLE_NAME>"
            }
 }
```

NOTE
<IAM_ROLE_NAME>: IAM name created previously. <your_aws_account_id>: It can be found in the top-right side of AWS console while selecting account name.

6. In the **Filter by type** drop down select **Customer managed** and select the previously created policy.



Permissions policies (1/1232)

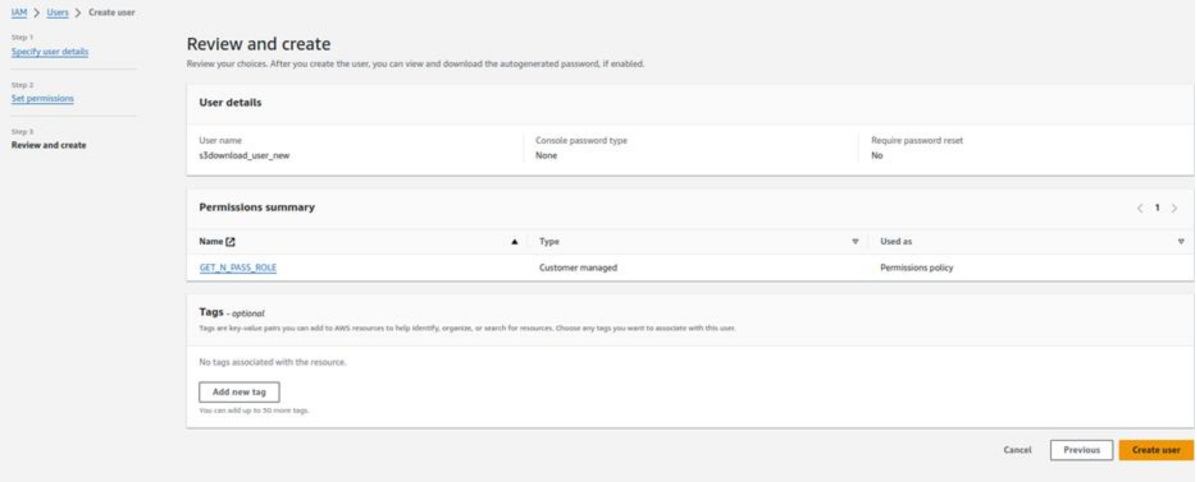
Choose one or more policies to attach to your new user.

Filter by Type: Customer managed ▼ — 7 matches

	Policy name	Type	Attached entities
□	aws-iot-rule-DemoRule-action-1-role-Demo_IAM_Role	Customer managed	1
☑	GET_N_PASS_ROLE	Customer managed	0
□	Getobjs3	Customer managed	3
□	passrolepermission	Customer managed	2
□	policy_aws_cli_permission	Customer managed	1
□	S3Getobjectpolicy	Customer managed	1

Figure 113. Selecting policy for user

7. Also select **AWSIotFullAccess** policy in **AWS managed** filter.
8. Select **Next**.
9. Select **Create User**.



IAM > Users > Create user

Step 1 Specify user details

Step 2 Set permissions

Step 3 Review and create

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name	Console password type	Require password reset
s3download_user_new	None	No

Permissions summary

Name	Type	Used as
GET_N_PASS_ROLE	Customer managed	Permissions policy

Tags - *optional*

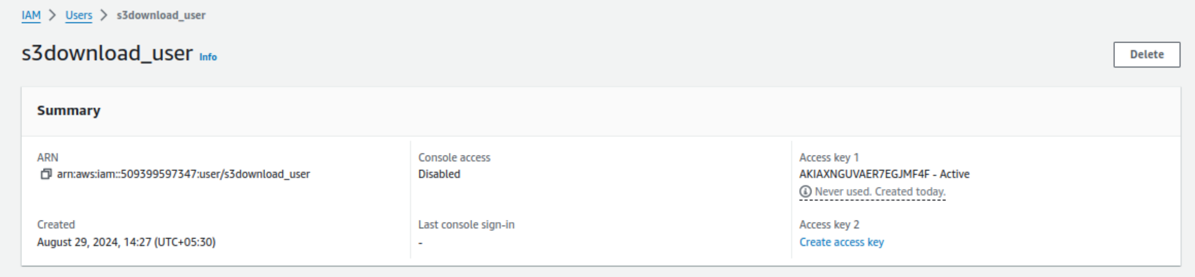
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

You can add up to 50 more tags.

Figure 114. Review and create user

10. Select **created user**.
11. Select **Create access key.**



IAM > Users > s3download_user

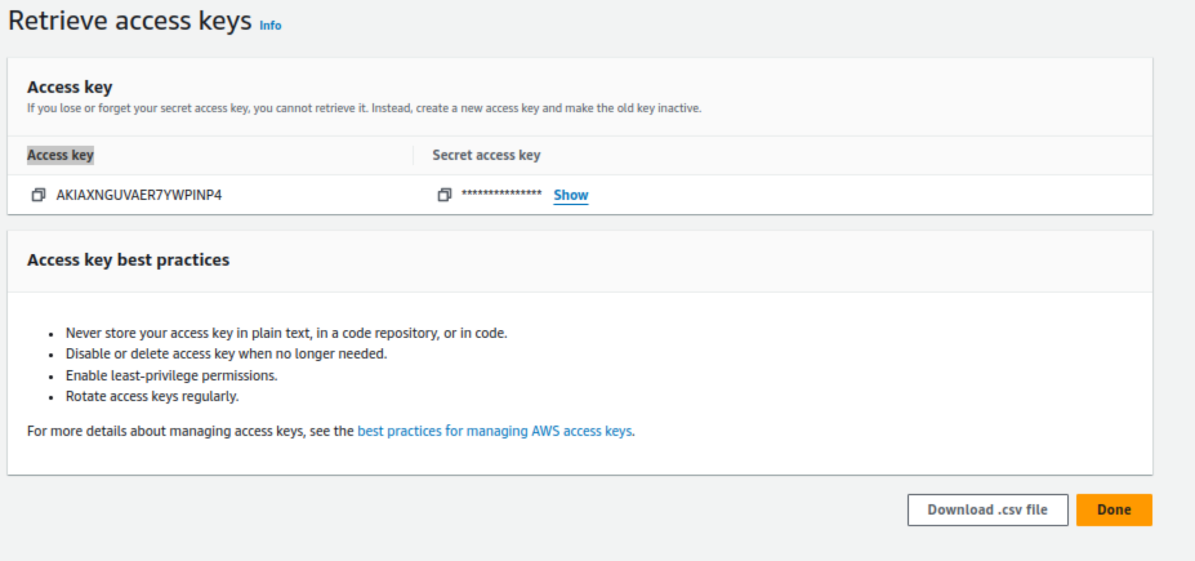
s3download_user Info

Summary

ARN arn:aws:iam::509399597347:user/s3download_user	Console access Disabled	Access key 1 AKIAXNGUVAER7EGJMF4F - Active Never used. Created today.
Created August 29, 2024, 14:27 (UTC+05:30)	Last console sign-in -	Access key 2 Create access key

Figure 115. Create access key

12. Select **CLI,** tick confirmation and choose **Next**.
13. (Optional) give description tag and select **Create access key**.
14. Download .csv file for storing **Access key** and **Secret access key**.



Retrieve access keys Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIAXNGUVAER7YWPINP4	*************** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the best practices for managing AWS access keys.

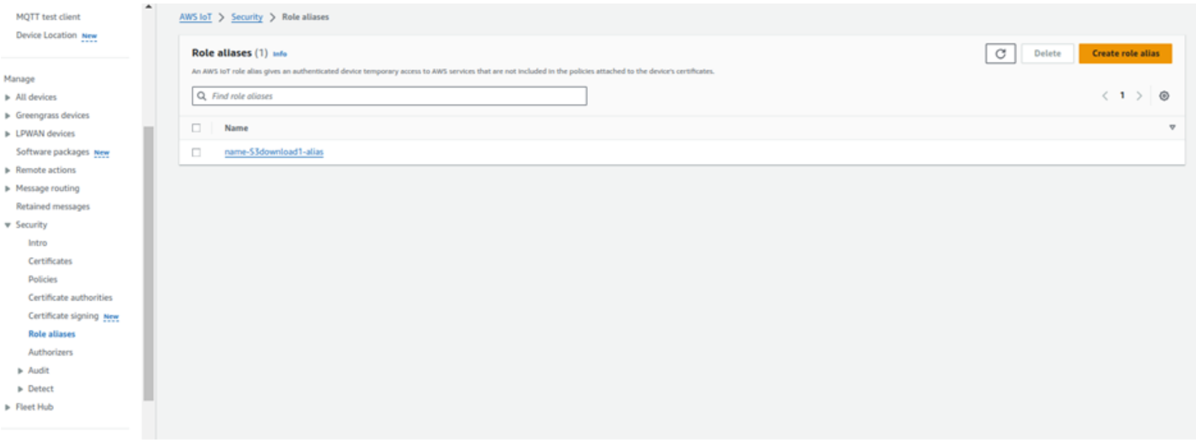
Figure 116. Retrieve access key

15. Select **Done**.

9.7 Create Role Alias

Now that you have configured the IAM role, you can create a role alias with AWS IoT. You must provide the following pieces of information when creating a role alias.

1. In the IoT console select **Security** > **Role aliases**.



Role aliases (1) Info

An AWS IoT role alias gives an authenticated device temporary access to AWS services that are not included in the policies attached to the device's certificates.

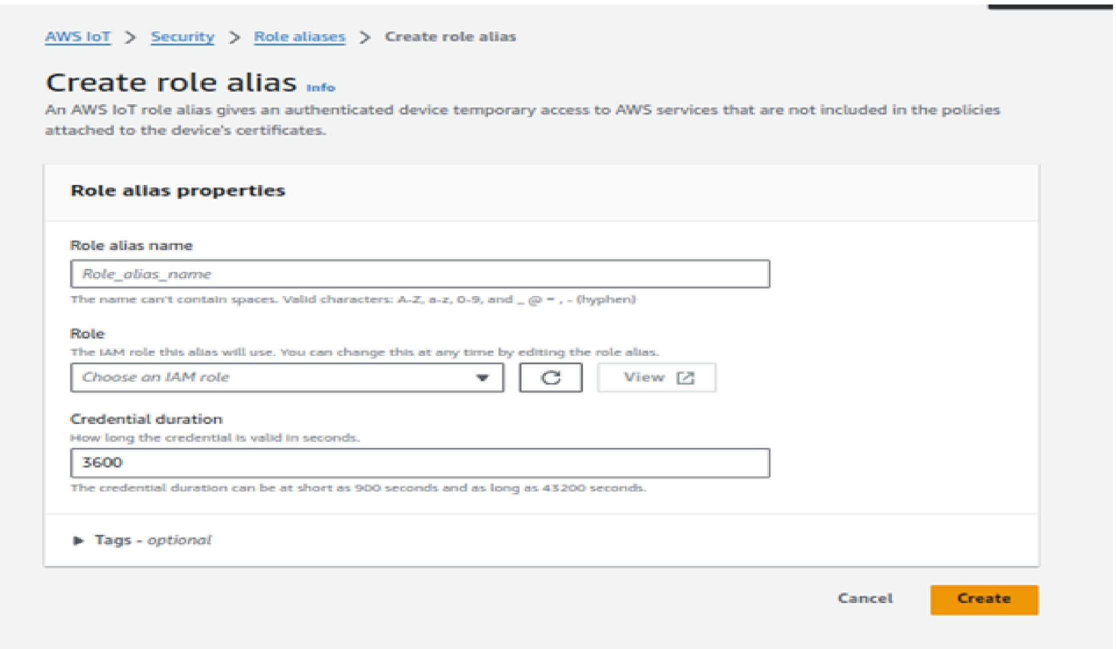
Find role aliases

□	Name
□	name-S3download1-alias

Figure 117. Create role alias

2. Select **Create role alias**.
3. Type **alias name**, **credential duration** and select **Role created** in Section 9.5.

NOTE
CredentialDurationSeconds: This is an optional attribute specifying the validity (in seconds) of the security token. The minimum value is 900 seconds (15 minutes), and the maximum value is 3,600 seconds (60 minutes).



AWS IoT > Security > Role aliases > Create role alias

Create role alias Info

An AWS IoT role alias gives an authenticated device temporary access to AWS services that are not included in the policies attached to the device's certificates.

Role alias properties

Role alias name

Role_alias_name

The name can't contain spaces. Valid characters: A-Z, a-z, 0-9, and _ @ = , . - (hyphen)

Role

The IAM role this alias will use. You can change this at any time by editing the role alias.

Choose an IAM role

Credential duration

How long the credential is valid in seconds.

3600

The credential duration can be at short as 900 seconds and as long as 43200 seconds.

Figure 118. Role alias properties

4. Select **Create**.

9.8 Attach Role Alias Policy to Certificate

1. Create custom policy using the given JSON and follow the steps in 6.1.2.3.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": "iot:AssumeRoleWithCertificate",
        "Resource": "arn:aws:iot:<Region>:<your_aws_account_id>:rolealias/<name_of_role_alias>"
      }
    ]
  }
```

NOTE
< name_of_role_alias>: Name of the role alias created. <your_aws_account_id>: It can be found in the top-right side of AWS console while selecting account name. <Region>: It can be selected in AWS console.

2. Attach the above created policy to certificate by following the steps in Section 6.1.2.4.

9.9 Request Security Token

1. To create or describe Endpoint, you need to install AWS CLI. To install it, follow the steps in the given link.
   https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html
2. After installing AWS CLI, you need to run the following command.

```
aws configure
```

3. Provide **aws_access_key_id** and **aws_secret_access_key** which generated in Section 9.6.
   Also, the region and output format (JSON).
4. Run the following command for getting security token after setting AWS CLI.

```
aws iot describe-endpoint --endpoint-type iot:CredentialProvider
```

5. The following **JSON** object is sample output of the describe-endpoint command. It contains the endpointAddress that you use to request a security token.

```
{
    "endpoint Address": "your_aws_account_specific_prefix.credentials.iot.your
region.amazonaws.com"
}
```

9.10 Code Configuration

Use the certificate, key, bucket created during the previous process to the macros in file **demo_config.h**.

```
#define democonfigIOT_CREDENTIAL_PROVIDER_ENDPOINT
"<your_aws_account_specific_prefix>.credentials.iot.us-east-1.amazonaws.com"

#define democonfigROOT_CA_PEM “root certificate downloaded when setting up the device
certificate in AWS Account Setup”
#define democonfigCLIENT_CERTIFICATE_PEM “client certificate downloaded when setting up the
device certificate in AWS IoT Account Setup"

#define  democonfigCLIENT_PRIVATE_KEY_PEM "private key downloaded when setting up the device
certificate in AWS IoT Account Setup"
```

```
#define democonfigTHING_NAME                          "Name of IOT Thing that you provided in STEP
1"
#define democonfigIOT_CREDENTIAL_PROVIDER_ROLE            "Name of ROLE ALIAS that you provided in
section 6.7"
#define democonfigS3_BUCKET_NAME                          "Name of Bucket that contains the object
that needs to be downloaded"
#define democonfigS3_BUCKET_REGION                        "Region where Bucket is located"
#define democonfigS3_OBJECT_NAME                          "Name of object that needs to be
downloaded from AWS S3"
```

9.11 Testing Demo

1. DA16x00 should be **enabled** with SNTP for using certificates and connected to AP with **internet** access.
2. After building the code, load the image in DA16x00.
3. The demo automatically downloads the s3 private bucket file content and displays it in the console.

```
DEBUG] [Demos] [prvSendS3HttpEmptyGet:1068] Received HTTP response from
s3privatebucketdemo.s3.us-east-1.amazonaws.com/test_s3_bucket.txt...
[DEBUG] [Demos] [prvSendS3HttpEmptyGet:1070] Response Headers:
x-amz-id-2: Py3YyB+XPyTCFH6dWE1WcyjY7q3Q/WBOZRxcqaybCSIumkbVv2oZ+LrpbyIhHK2w6443WuJt/D8=
x-amz-request-id: RVWBXMZQAKJ8FRWV
Date: Thu, 22 Aug 2024 19:02:12 GMT
Last-Modified: Thu, 22 Aug 2024 17:40:55 GMT
ETag: "2e5148afee6432823aee10a27a3a52ff"
x-amz-server-side-encryption: AES256
Accept-Ranges: bytes
Content-Range: bytes 0-1024/1025
Content-Type: text/plain
Server: AmazonS3
Content-Length: 1025
[DEBUG] [Demos] [prvSendS3HttpEmptyGet:1073] Response Body:
6612564665543818482403435750349017663609926499163346576220149801451912389185926873398365303938
87264326429951433585045690077771
5859869340249686694340283504163457022411806633040456823648322149407649291709884486624991429087
99298664245623314794704849295 30
4798107198075017717708753814435626352262734959756725609267280962722018526857388403754623314994
10484257218860173970024937 71038
```

```
5978949352294638874287215930948390792479864689759029679908713843203529304159229725861615620844
3607672462374144231313952523825
4121472243678952135750691080678438523913121266791528606569722357719234953663106981929185242016
1751071280762096700317526464632
9092876562122951842146119916941895931718937037709622303904807519784876983985859485514354675809
3458201630388955491473164903161
1902973368535645741909205082336233397713399375892739362196688036541411080980862571111620497249
4708604941468381375412202718800
3075727614346439528964487690991586649321220625005355040038529367337670153746836096076465791378
6708380781323834871961191069325
5294339716425075



[Sockets_Disconnect:449]
[INFO] [Demos] [prvHTTPDemoTask:706] Demo iteration 1 was successful.
[INFO] [Demos] [prvHTTPDemoTask:724] prvHTTPDemoTask() completed successfully. Total free heap
is 168608.

[INFO] [Demos] [prvHTTPDemoTask:727] Demo completed successfully.

[INFO] [Demos] [prvHTTPDemoTask:728] -------DEMO FINISHED-------
```

Appendix A Provisioning

The DA16200 supports a provisioning feature called Soft AP mode for an easy network configuration. Provisioning with the **mobile network data off** on your mobile phone and Wi-Fi turned on. When provisioning is complete, turn on your mobile data again. Figure 119 shows the workflow of the provisioning process.

- Press the **Factory Reset** button for about 5 seconds. Start the Android application and touch the **START** button to find the wanted AP.

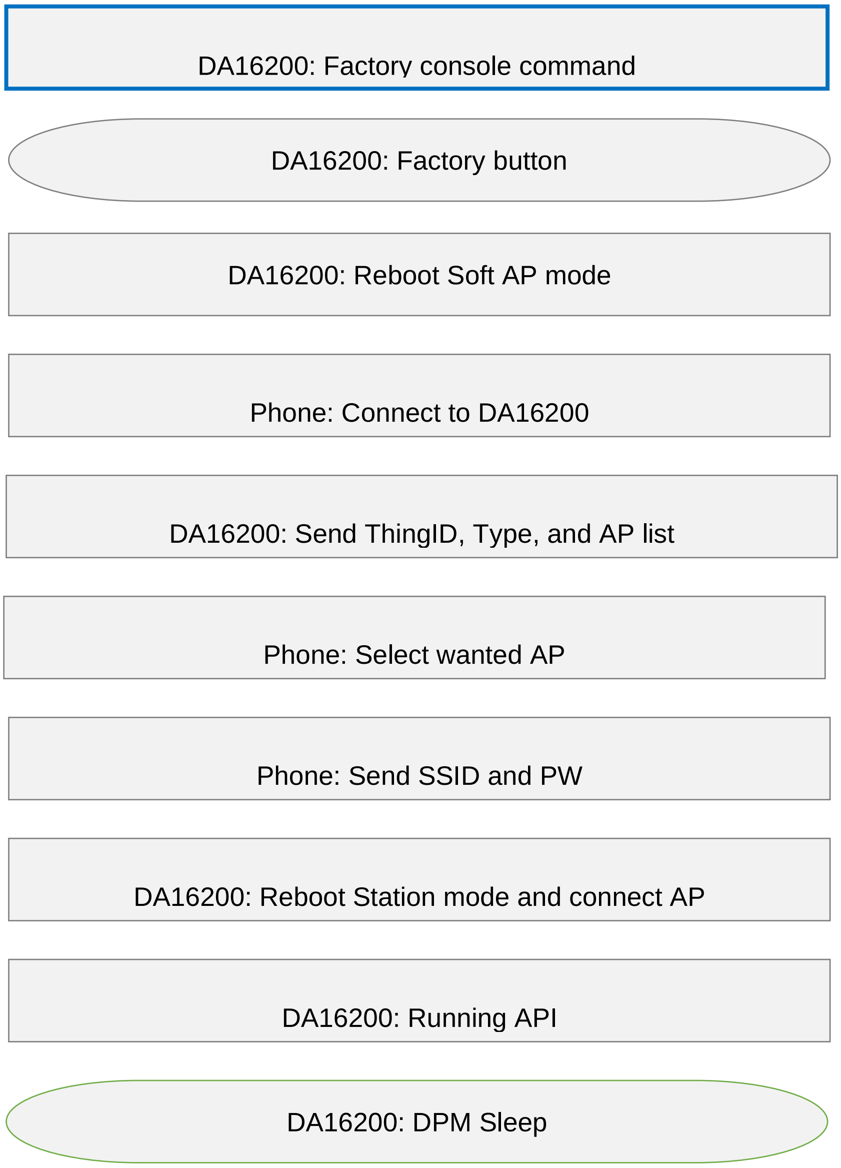
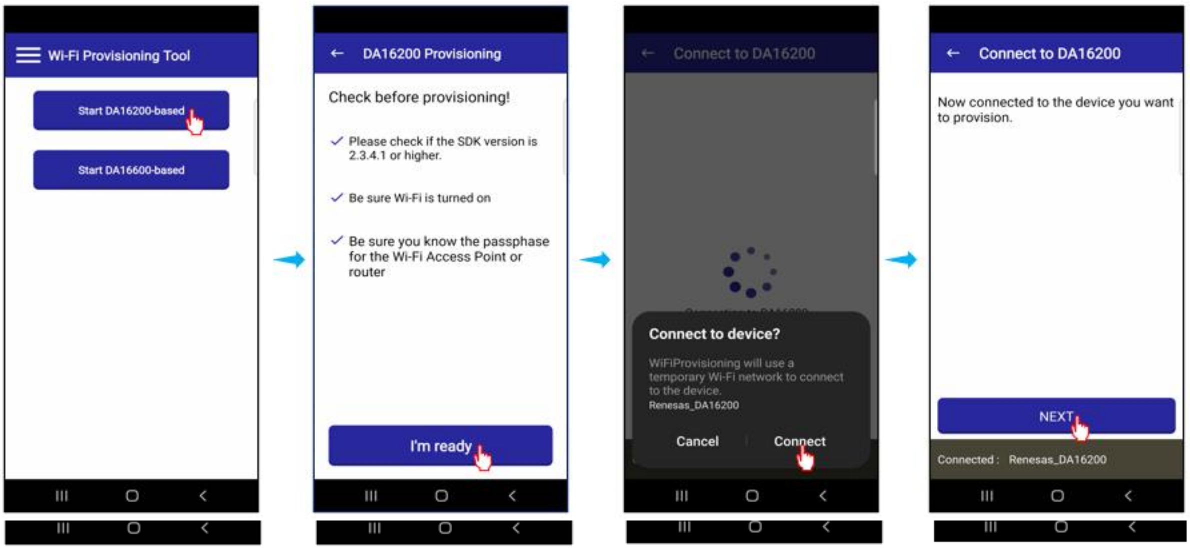

DA16200: Factory console command
DA16200: Factory button
DA16200: Reboot Soft AP mode
Phone: Connect to DA16200
DA16200: Send ThingID, Type, and AP list
Phone: Select wanted AP
Phone: Send SSID and PW
DA16200: Reboot Station mode and connect AP
DA16200: Running API
DA16200: DPM Sleep

Figure 119. Provisioning flow

A.1 Android Application

```
System Mode : Soft AP (1)

>>> DHCP Server Started
>>> Start DA16X Supplicant ...
>>> DA16x Supp Ver2.7 - 2022_03
>>> Add SoftAP Inteface (softap1) ...
>>> MAC address (softap1) : d4:3d:39:11:5e:73
>>> softap1 interface add OK
>>> AP Operating Channel: 1(2412)

>>> Network Interface (wlan1) : UP
BSS Isolate Disabled

Soft AP is Ready (d4:3d:39:11:5e:73)

===========================================
[ APP-IOT Doorlock ]
[ aws_shadow_dpm_auto_start]
AWS_IOT on Station Mode for "APP-DOORLOCK-1"
============================================

AWS_IOT AP Mode  APP-DOORLOCK-1

=======================================================
[Start Provisioning with TCP/TLS] .. Soft AP Mode
=======================================================

[app_provision_switch_client_thread] Create...(status=0) [10]
[app_provision_TCP_server_thread] Create ...
[app_provision_TLS_server_thread] Create TLS...

>>> Start Provisioning Server (TLS) ...
Wait Accept (TLS)...

> Wi-Fi Scan request success.
(0) KT_GIGA_2G_505 / 3 / -25 / 2412
(1) TP-LINK_AECC / 3 / -40 / 2412

[app_provision_TCP_server_thread] socket().. status=1
 Wait Accept...
```



Check before provisioning!

✓ Please check if the SDK version is 2.3.4.1 or higher.

✓ Be sure Wi-Fi is turned on

✓ Be sure you know the passphase for the Wi-Fi Access Point or router

Connect to device?

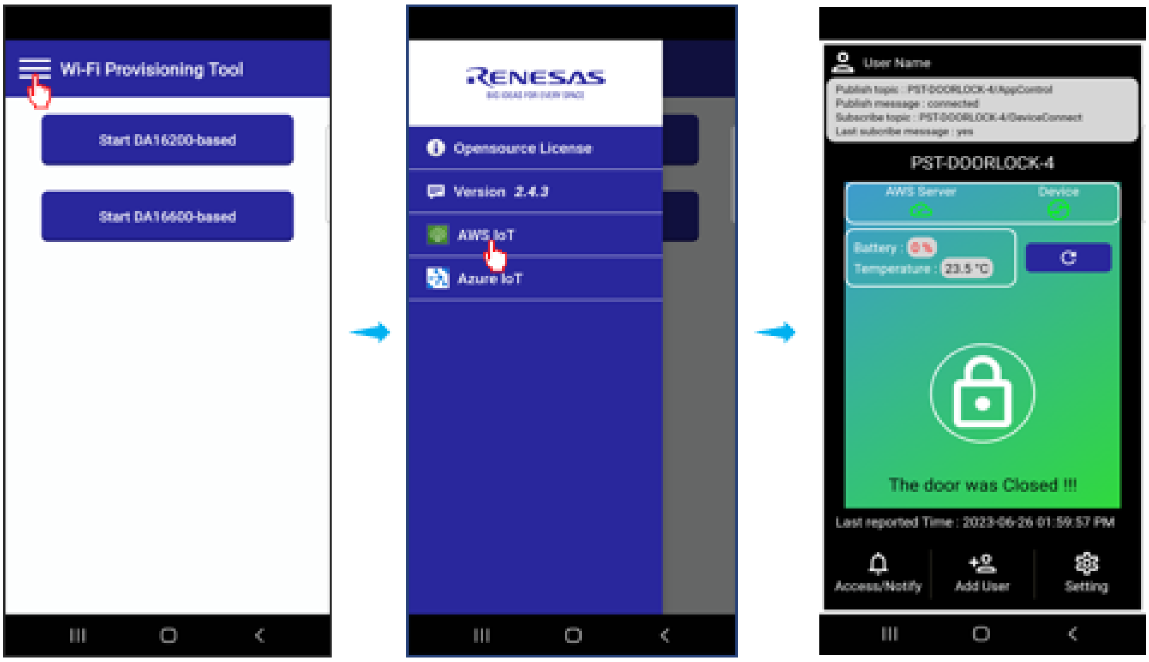
WiFiProvisioning will use a temporary Wi-Fi network to connect to the device.
Renesas_DA16200

Now connected to the device you want to provision.

Figure 120. Provisioning from mobile app

```
[dpmAPPManager] DM_NEED_CONNECTION

DM_NEED_CONNECTION


[INFO] [DoorLockDemo] [aws_dpm_app_connect:2267] Establishing MQTT session with provisioned
certificate...

recv timeout(=2000 ms) set OK (socket=0)

hostName = "a1kzdt4nun8bnh-ats.iot.ap-northeast-1.amazonaws.com", flag to re-query (=0)

host IP = "52.69.14.255"

TCP connection OK to "a1kzdt4nun8bnh-ats.iot.ap-northeast-1.amazonaws.com"

recv timeout(=120 ms) set OK (socket=0)

[INFO] [DoorLockDemo] [aws_dpm_app_connect:2317] Sucessfully established connection with provisioned
credentials.

[Make AWS-Thing-Name]

[NVRAM] AWS Thing name : [APP-DOORLOCK-1] (len=14)

[NVRAM] [APP-DOORLOCK-1/DeviceConnect][APP-DOORLOCK-1/AppControl][APP-DOORLOCK-1/DeviceControl]

[INFO] [DoorLockDemo] [aws_dpm_app_subscription:1939] subscription info: total(default:4, tried:4), OK(4)

current RTM user Timer ID = 0

current RTM temperature(str): 0.000000

current RTM battery(str): 0.000000

current RTM doorOpen state: "false"

current RTM doorOpenMode : 0

current RTM FOTAFlag: 0

current RTM FOTA url : ""

[dpmAPPManager] DM_BOOT_WAKEUP

DM_WAKEUP_BOOT


[INFO] [DoorLockDemo] [connectionReadyInform:1598] publish (command response) OK -  payload: "yes"

[closeControl]
```

```
=======================================================================================
[INFO] [DoorLockDemo] [aws_dpm_app_door_work:2030] publish (shadow doorlock update) OK -        payload: 
"{"state":{"reported":{"doorState":false,"openMethod":"app","doorStateChange":1,"doorOpenMode":0,"OTAupdate":0,"OTAresult":"OTA_UNKNOWN"}}}"
*****************************************************************************************


last user Timer ID = 0
last doorOpenFlag state: "false"
last FOTA Stat: 0
last FOTA Url: ""


Sleep mode 3: KA timer interval(=1800 sec)


DM_FINISH_DEVICE


recv timeout(=20 ms) set OK (socket=0)
[dpm_keepalive_timer_register] RTC interval (=1780 secs), mode (=0)
>>> Start DPM Power-Down !!!
```



Publish topic : PST-DOORLOCK-4/AppControl
Publish message : connected
Subscribe topic : PST-DOORLOCK-4/DeviceConnect
Last subcribe message : yes

The door was Closed !!!

Last reported Time : 2023-06-26 01:59:57 PM

Figure 121. Running AWS IoT application from mobile app

Appendix B AT Commands for AWS IoT

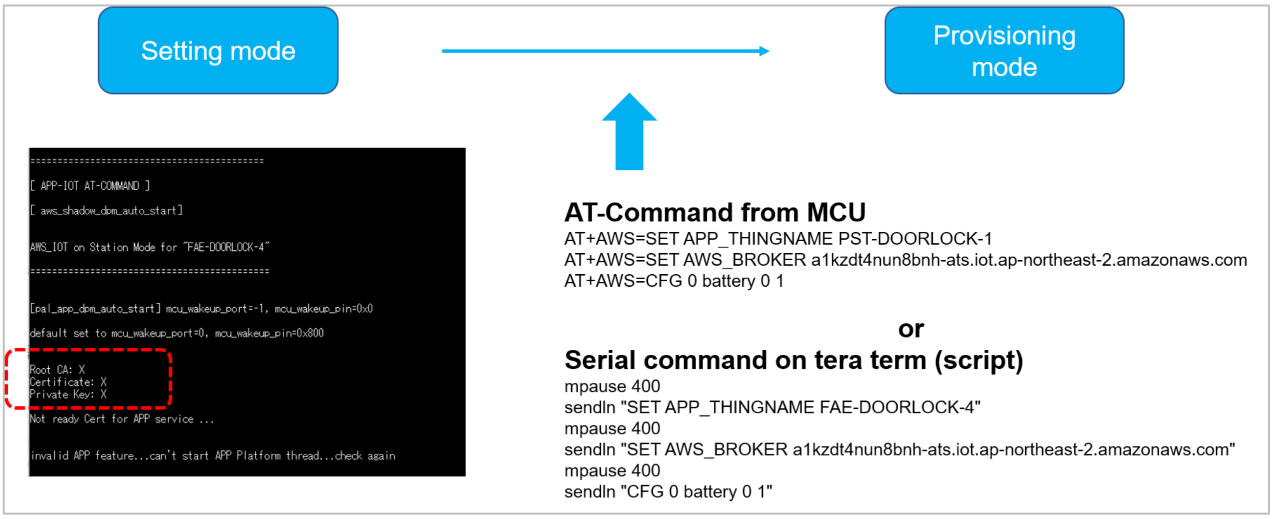
B.1 Operating Modes

There are three operating modes:

- Setting Mode for features configuration.
- Provisioning Mode for network connection.
- Communication Mode for running.

B.1.1 Setting Mode

After uploading the image and rebooting, the DA16200/DA16600 enters Setting mode. In this mode, all AWS IoT settings can be configured using the SET command and a specific topic can be configured using the CFG command. For proper operation of AWS IoT, the TLS certificate keys must be set. All configuration data is stored before calling the factory reset command (Figure 122).



```
==========================================
[ APP-IOT AT-COMMAND ]
[ aws_shadow_dpm_auto_start]

AWS_IOT on Station Mode for "FAE-DOORLOCK-4"
===========================================

[pal_app_dpm_auto_start] mcu_wakeup_port=-1, mcu_wakeup_pin=0x0
default set to mcu_wakeup_port=0, mcu_wakeup_pin=0x800

Root CA: X
Certificate: X
Private Key: X
Not ready Cert for APP service ...

invalid APP feature...can't start APP Platform thread...check again
```

AT-Command from MCU

```
AT+AWS=SET APP_THINGNAME PST-DOORLOCK-1
AT+AWS=SET AWS_BROKER a1kzdt4nun8bnh-ats.iot.ap-northeast-2.amazonaws.com
AT+AWS=CFG 0 battery 0 1
```

or

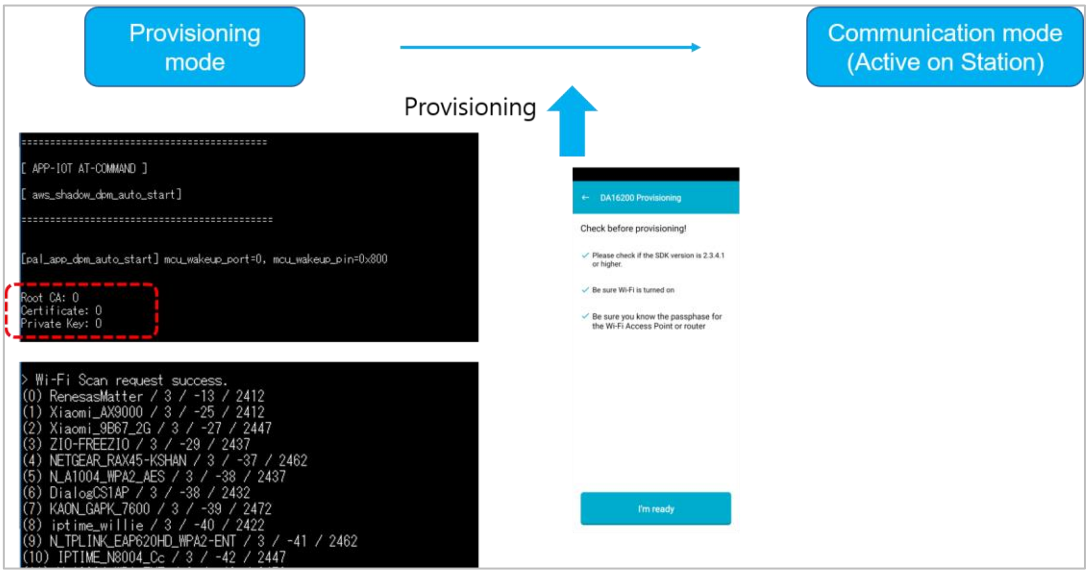
Serial command on tera term (script)

```
mpause 400
sendln "SET APP_THINGNAME FAE-DOORLOCK-4"
mpause 400
sendln "SET AWS_BROKER a1kzdt4nun8bnh-ats.iot.ap-northeast-2.amazonaws.com"
mpause 400
sendln "CFG 0 battery 0 1"
```

Figure 122. Setting mode

B.2 Provisioning Mode

In provisioning mode, the DA16200/DA16600 can be provisioned using an Android or iOS device. During provisioning, the MCU only receives a report on the provisioning status. When provisioning is complete, the DA16200/DA16600 enters Communication mode automatically after rebooting (Figure 123).



```
============================================
[ APP-IOT AT-COMMAND ]
[ aws_shadow_dpm_auto_start]
============================================

[pal_app_dpm_auto_start] mcu_wakeup_port=0, mcu_wakeup_pin=0x800

Root CA: 0
Certificate: 0
Private Key: 0
```

```
> Wi-Fi Scan request success.
(0) RenesasMatter / 3 / -13 / 2412
(1) Xiaomi_AX9000 / 3 / -25 / 2412
(2) Xiaomi_9B67_2G / 3 / -27 / 2447
(3) ZIO-FREEZIO / 3 / -29 / 2437
(4) NETGEAR_RAX45-KSHAN / 3 / -37 / 2462
(5) N_A1004_WPA2_AES / 3 / -38 / 2437
(6) DialogCS1AP / 3 / -38 / 2432
(7) KAON_GAPK_7600 / 3 / -39 / 2472
(8) iptime_willie / 3 / -40 / 2422
(9) N_TPLINK_EAP620HD_WPA2-ENT / 3 / -41 / 2462
(10) IPTIME_N8004_Cc / 3 / -42 / 2447
```

DA16200 Provisioning

Check before provisioning!

✓ Please check if the SDK version is 2.3.4.1 or higher.

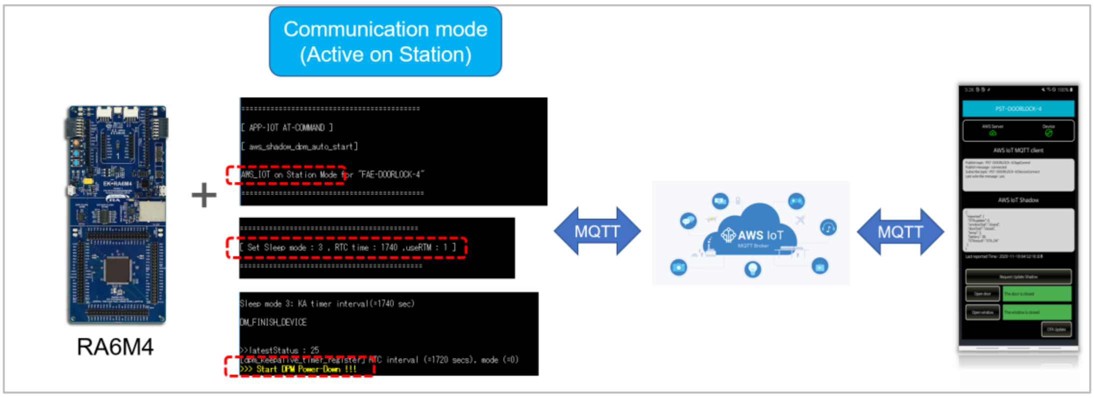
✓ Be sure Wi-Fi is turned on

✓ Be sure you know the passphase for the Wi-Fi Access Point or router

Figure 123. Provisioning mode

B.2.1 Communication Mode

The DA16200/DA16600 Communication Mode is used by the MCU to communicate (send and receive) topic values with an AWS server (Figure 124).



```
============================================
[ APP-IOT AT-COMMAND ]
[ aws_shadow_dpm_auto_start]

AWS_IOT on Station Mode for "FAE-DOORLOCK-4"
============================================
```

```
============================================
[ Set Sleep mode : 3 , RTC time : 1740 ,useRTM : 1 ]
============================================
```

```
Sleep mode 3: KA timer interval(=1740 sec)
DM_FINISH_DEVICE

>>latestStatus : 25
[dpm_keepalive_timer_register] RTC interval (=1720 secs), mode (=0)
>>> Start DPM Power-Down !!!
```

Figure 124. Communication mode

B.3 Configuring Topic to Publish, Subscribe, and Shadow

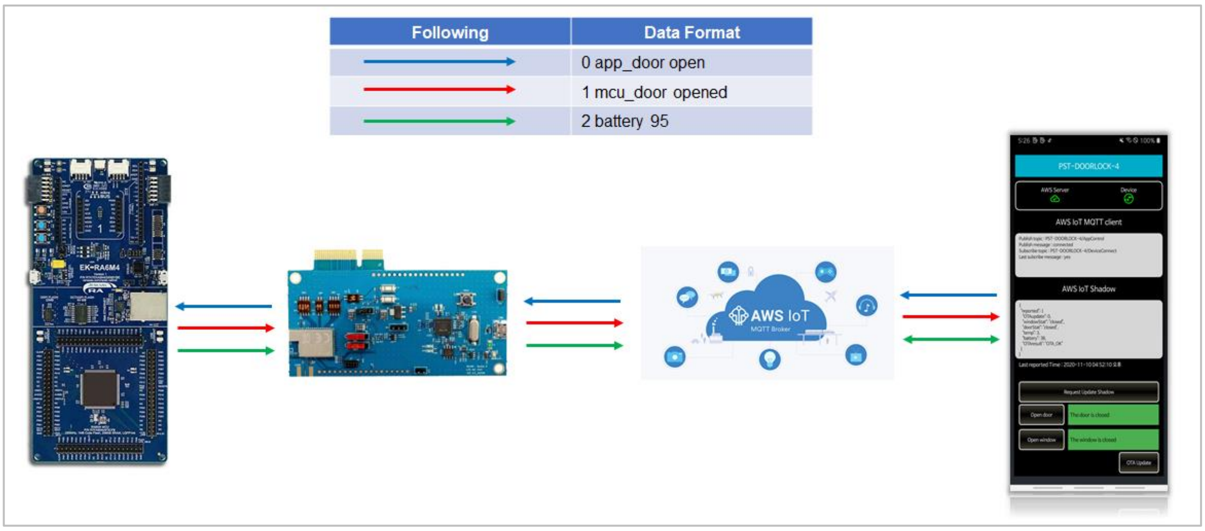
B.3.1 Configure Topics

- Topics are configured as shown in Table 6.
- The MCU and Mobile App should be configured based on the topics shown in Table 6.
- The MCU pushes the topics in Table 6 to the DA16200/DA16600 using AT command.

The DA16200 facilitates the communication between the MCU and phone as shown in Figure 125.

Table 6. Configuration of topics

Number	Name	Value type	CMD type	Value
0	app_door	1: String	2: Subscribe	"open"/"close"
1	mcu_door	1: String	0: Publish	"opened"/"closed"
2	battery	0: Integer	1: Shadow	Battery value (0~100)
3	temperature	2: Float	1: Shadow	Temperature value
4	doorStat	1: String	1: Shadow	"opened"/"closed"
5	windowStat	1: String	1: Shadow	"opened"/"closed"



Following	Data Format
→	0 app_door open
→	1 mcu_door opened
→	2 battery 95

Figure 125. Communication between MCU and phone

B.4 AT Command List

B.4.1 Basic Set

Table 7. Basic set of MCU to DA16200/DA16600

Head	Main	Sub	Parameters
AT+AWS=	SET	APP_THINGNAME	Set the device thing name. Used to choose a device by its thing name during provisioning.
		AWS_BROKER	Set the broker address.
		APP_LPORT	Set the local port.
		APP_SUBTOPIC	Set subscriber topic name, and the default is "/AppControl".
		APP_PUBTOPIC	Set subs topic name, and the default is "/DeviceControl".
		SLEEP_MODE	Set sleep mode. 1 – not connected sleep. The DA16200/DA16600 wakes up only by RTC_PWR_KEY. 2 – not connected sleep. The DA16200/DA16600 wakes up by RTC. 3 – connected sleep. The connection is retained even during DPM.
		USE_DPM	Define the operation of sleep mode 3. 0 – no DPM. Used during debugging. 1 – DPM mode.
		RTC_TIME	Set the wake-up time for Sleep mode 2.
		DPM_KEEP_ALIVE	Set the keep-alive time between the IoT device and the AP. Default value is 30*1000 microseconds.
		USE_WAKE_UP	Set the wake-up time for full-boot mode. Default value is set to 0 (0 = unused).
		TIM_WAKE_UP	Set the period to check a beacon frame from the AP. Default value is set to10.
		AWS_USE_FP	Not used command. 0 – Default value. 1 – Not in use.
For example: AT+AWS=SET APP_THINGNAME AssignedThingName AT+AWS=SET AWS_BROKER a1kzdt4nun8bnh-ats.iot.ap-northeast-2.amazonaws.com			

B.4.2 TLS Certificate

Table 8. TLS from MCU to DA16200/DA16600

Start code	Sub code	Type	End code
\x1b	C0,	Root CA. Self-Signed, well known. Has root certificate public key. Signed by root certificate private key.	\x03
	C1,	Certificate key. Has own public key. Signed by root certificate private key. Use root certificate public key to prove authenticity.	
	C2,	Private key. Has own public key. Signed by certificate private key. Use certificate 1 public key to prove authenticity.	
For example: send "\x1b" over UART send "C0,-----BEGIN CERTIFICATE-----\n" "MIIDQTCCAimgAwIBAgITBmyfz5m/jAo ….. over UART send "\x03"			

B.4.3 PIN MUX

Table 9. PIN MUX from MCU to DA16200/DA16600

Head	Main	Sub	Parameters
AT+AWS=	SET	NV_PIN_AMUX	AMUX_UART1d 4 /* UART1(RXD, TXD) */ AMUX_GPIO 9 /* GPIOA [1:0] */
		NV_PIN_BMUX	BMUX_UART1d 4 /* UART1(RXD, TXD) */ BMUX_GPIO 8 /* GPIOA [3:2] */
		NV_PIN_CMUX	CMUX_UART1d 6 /* UART1(RXD, TXD) */ CMUX_GPIO 8 /* GPIOA [5:4] */
		NV_PIN_DMUX	DMUX_UART1d 4 /* UART1(RXD, TXD) */ DMUX_GPIO 8 /* GPIOA [7:6] */
		NV_PIN_EMUX	EMUX_GPIO 8 /* GPIOA [9:8] */
		NV_PIN_FMUX	FMUX_GPIO 6 /* GPIOA [11:10] */
		NV_PIN_UMUX	UMUX_GPIO 2 /* GPIOC [8:6] */
		APP_MCU_WKAEUP_PORT	GPIO_UNIT_A 0 GPIO_UNIT_C 2 /*Support only GPIO 6,7,8 */
		APP_MCU_WKAEUP_PIN	GPIO_PIN0 ~ GPIO_PIN11
		UART_CFG	[baud-rate]
Note: Default pin mux is BMUX For example: use GPIOA2 and GPIOA3 for UART1, and GPIOA9 for MCU wakeup AT+AWS=SET NV_PIN_BMUX BMUX_UART1d AT+AWS=SET NV_PIN_EMUX EMUX_GPIO AT+AWS=SET APP_MCU_WKAEUP_PORT GPIO_UNIT_A AT+AWS=SET APP_MCU_WKAEUP_PIN GPIO_PIN9			

B.4.4 Configure Data as Topics

Table 10. Configuration data from MCU to DA16200/DA16600

Head	Main	Sub	Parameters
AT+AWS=	CFG	[number] [name] [value-type] [MQTT-type]	▪ number: Index to identify the saved topic. Increase by 1 when setting a new topic. Max value is 10 (total supported topics is 10). ▪ name: String specifying the topic name. ▪ value-type 0 – Integer type. 1 – String type. 2 – Float type. ▪ MQTT-type 0 – Publish: The prompt command is used to send a value from the MCU to the phone. For example, door state = true/false. 1 – Shadow: The value is sent to the device twin and is updated on the phone the next time it is connected. 2 – Subscribe: The prompt command is used to send a value from the phone to the MCU. For example, door open command.
For example: AT+AWS=CFG 0 doorStat 1 1 AT+AWS=CFG 1 battery 2 1 AT+AWS=CFG 2 door_open 0 2			

B.4.5 Command – MCU to DA16200/DA16600

Table 11. Command of MCU to DA16200/DA16600

Head	Main	Sub	Description
AT+AWS=	CMD	FACTORY_RESET	Reset the AWS IoT configuration to the factory default. All values stored in NVRAM are cleared. Use the "SET" and "CFG" commands to set the AWS IoT configuration.
		RESET_TO_AP	Switch to AP mode keeping the values set in NVRAM. The previous values in NVRAM are kept.
		GET_STATUS	Get the current AWS IoT status. The MCU can read the current status from the DA16200/DA16600 at any time.
		RESTART	Reboot the device keeping the current mode and status.
		MCU_DATA	Used by the MCU to set a CFG parameter in the DA16200/DA16600.The value must be the same format as defined by the CFG setting. Parameters: [number] [name] [value]
For example: AT+AZU=CMD FACTORY_RESET AT+AZU=CMD MCU_DATA 1 mcu_door opened			

B.4.6 Command – DA16200/DA16600 to MCU

Table 12. Command of DA16200/DA16600 to MCU

Head	Main	Parameters	Description
+AWSIOT	SERVER_DATA	[number] [name] [value]	Used by the DA16200/DA16600 to set a CFG parameter in the MCU. The value must be the same format as defined by the CFG setting.
+AWSIOT	CMD_TO_MCU	update	Used by the DA16200/DA16600 to request the status of devices such as sensors, batteries, and doors from the MCU. The DA16200/DA16600 maintains the values obtained from the MCU and forwards them when requested by an external phone app or by an MQTT ping-pong wake-up event.
For example: +AWSIOT SERVER_DATA 0 door_control open +AWSIOT CMD_TO_MCU update			

B.4.7 DA16200/DA16600 Status – DA16200/DA16600 to MCU

Table 13. Status from DA16200/DA16600 to MCU

Status	Value	Parameters
IDLE	-1	Initial state of AWS-IoT application. Sent when a system error occurs. For example, network connection failure.
Done factory reset	0	Sent after completes factory reset by "CMD FACTORY_RESET".
Boot Ready	1	Sent when entering AWS-IoT application mode.
Need configuration	5	Sent if there is no setting. MCU should set and configure with the SET and CFG command.
Start AP mode	10	Sent when being started to AP mode. Need to process provisioning with Phone.
Network OK	15	Sent when it is OK to connect AP without problem.
Network fail	16	Sent when it fails to connect AP with any problem. Normally, it happens during provisioning failure by the wrong SSID or PW. Need to go to AP mode by MCU send "RESET_TO_AP" command.
Start STA	20	Not defined yet.
Done STA	25	Sent when entering Sleep mode for DPM.
MCUOTA	30	Sent when MCU OTA starts processing.
For example: +AWSIOT STATUS 15		

Appendix C Troubleshooting

C.1 Operational Issue

When UI buttons are not visible or not showing up properly while using the mobile app, try to uninstall and install the app again. The first time running the mobile app after reinstalling it, make sure that the app can access the location of the device as described in Test Provisioning on Android/iPhone Sections of Ref. [4].

10. Revision History

Revision	Date	Description
1.7	Oct 7, 2024	▪ Added Section 3, 4, 5. ▪ Added URL link in Section 2. ▪ Added Section 9.
1.6	July 22, 2024	▪ Modified Note to provide customer with thing name for testing instead of providing the AWS login credentials. ▪ Added Section 4.1. ▪ Editorial changes.
1.5	Jan 26, 2024	Added Troubleshooting Section.
1.4	Nov 30, 2023	Merged documents: ▪ UM-WI-016 DA16200 Door Lock Application Using AWS IoT. ▪ UM-WI-017 DA16200 AWS IoT Server Setup. ▪ UM-WI-038 DA16200 DA16600 Getting Started with AWS IoT Using AT Commands.
1.3	Aug 18, 2023	▪ Changed IDE to e2studio. ▪ Editorial update.
1.2	Dec 1, 2022	Edited as direct link of documents.
1.1	Nov 4, 2022	Modify hyperlink of the documents.
1.0	Oct 13, 2022	Initial version.

Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.

RoHS Compliance

Renesas Electronics’ suppliers certify that its products are in compliance with the requirements of Directive 2011/65/EU of the European Parliament on the restriction of the use of certain hazardous substances in electrical and electronic equipment. RoHS certificates from our suppliers are available on request.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.