

User Guide

SSK Cloud Connect

Date: November 23, 2020 | Version 1.0
FINAL



The Solutions People



CONTENTS

1	INTRODUCTION	3
1.1	Purpose of the Document	3
1.2	Prerequisite	3
2	IAM USER	4
2.1	Description	4
2.2	Creating IAM User	4
2.3	Listing IAM User	5
3	DEVICE	6
3.1	Description	6
3.2	Add a Device	6
3.3	Listing Devices.....	7
3.4	Deny Device	7
3.5	Allow Device	8
4	GREENGRASS	9
4.1	Description	9
4.2	Creating a Green grass Group.....	9
4.3	Listing Green grass Group	10
4.4	Green grass Subscription.....	10
4.5	Green grass Device	12
4.6	Green grass Deployment.....	13
5	OTA UPDATES	15
5.1	Description	15
5.2	Create a FreeRTOS OTA update job (Schedule OTA)	15
5.3	Create Custom Job (Schedule Job)	16
5.4	Listing OTA UPDATES	17
5.5	Create a Code Signing profile.....	18
6	CERTIFICATE.....	19
6.1	Description	19
6.2	Register CA	19
6.3	Listing Certificates	20
6.4	Add a Certificate.....	21
6.5	Create a Certificate (One Click)	23
7	POLICY.....	24
7.1	Description	24
7.2	Creating a Policy.....	24
7.3	Listing the Policy.....	25
7.4	Attach Policy	26
7.5	Detach Policy	26
8	REFERENCES	27

1 INTRODUCTION

1.1 Purpose of the Document

The SSK cloud connect guide provides an overview of the application to setup IAM User, Devices, AWS Greengrass, Certificate and Policies.

1.2 Prerequisite

For AWS Cloud Services descriptions and its background information, follow the [SSK Cloud Connect Installation Setup Guide v1.0 - FINAL](#)

To login SSK portal please refer section 3(step:19) from [SSK Cloud Connect Quick Start Guide v1.0 FINAL](#)

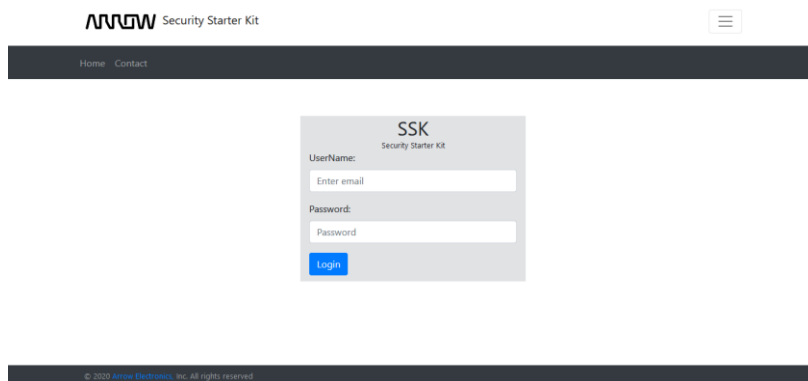


Figure 1: Login page

Note: AWS cloud does not support sending emails to an account tied address at the 'user' IAM level, due to that the 'Forgot Password' operation is not available.

2 IAM USER

2.1 Description

An AWS Identity and Access Management (IAM) user is an entity that you create in SSK Cloud connect to represent the person or application that uses it to interact with SSK Cloud connect. A user in SSK Cloud connect consists of a name and password.

2.2 Creating IAM User

An IAM user is a resource in IAM that has associated credentials and permissions. An IAM user can represent a person or an application that uses its credentials to make SSK Cloud connect requests.

Figure 2: Add IAM User

An IAM user can be described by the following:

1. **First Name**
The first name of the user with min 4 characters.
2. **Last Name**
The last name of the user.
3. **User Name**
The unique name of the user with min 4 characters. The User Name is used to log in to the SSK site/system. It should be unique across your AWS account.
4. **Password**
The password of the user. At least 16 chars with combination of one Upper case, Lower case, Number, and Special character.

5. Email Id

The Email Id of the user. Does not need to be unique.

6. Mobile Number

The Mobile number of the user with min 8 chars.

7. Permission

- a. **User** - can access only dashboard access and demo page
- b. **Tech Support** - can manage device, group, OTA, certificate, policy
- c. **Admin** - can manage all functionality.

8. Group Name

Upon selection of group name, user will be added in AWS Group to manage access

2.3 Listing I-AM User

Once you create ^aAn IAM user, you can ~~list down an IAM user at~~ view that user and any other users you have created on the List User page. Which is shown hHere in below images.

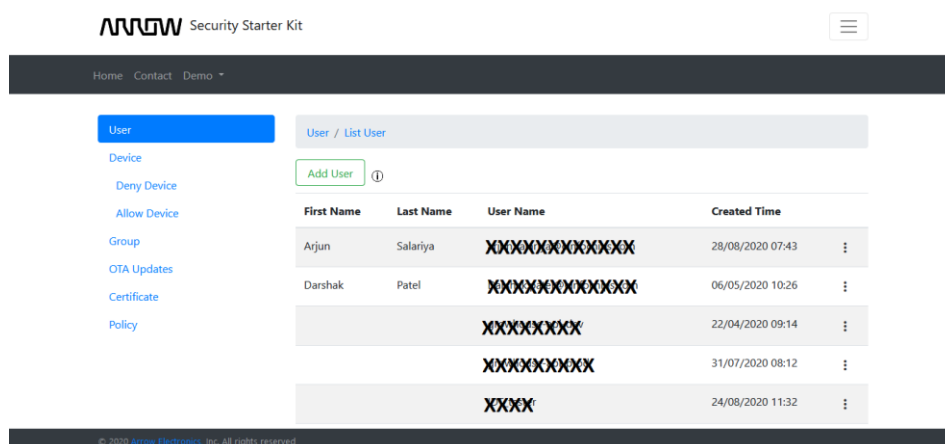


Figure 2: Listing I AM User

3 DEVICE

3.1 Description

AWS IoT provides a device registry that helps you manage your devices. A device is the representation of a physical device or logical entity. It can be a physical device or sensor (for example, a light bulb or a switch on a wall). It can also be a logical entity like an instance of an application or physical entity that does not connect to AWS IoT, but is related to devices that do (for example, a car that has engine sensors or a control panel).

Devices are identified by a name. Devices can also have attributes, which are name-value pairs you can use to store information about the device, such as its serial number or manufacturer. Adding your devices to the device registry allows you to manage and search for them more easily.

3.2 Add a Device

A device is the representation of a physical device or logical entity in the cloud. A device can be described by the following:

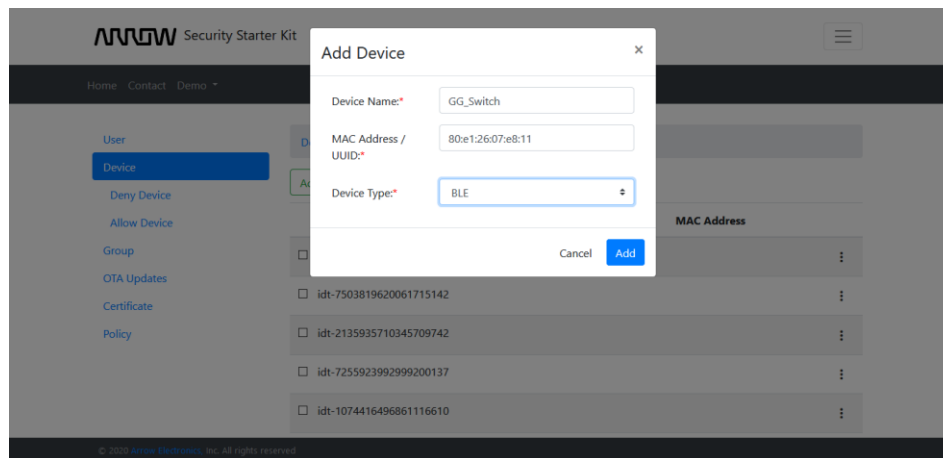


Figure 1: Add a device

1. **Device Name**
The name of the device.
2. **MAC Address**
The MAC Address of the device.

3. Device Type

Device types allow you to store description and configuration information that is common to all devices associated with the same device type. i.e. BLE, LTE-M, WIFI

3.3 Listing Devices

Listing of Devices can be seen in following figure with path SSK -> Device.

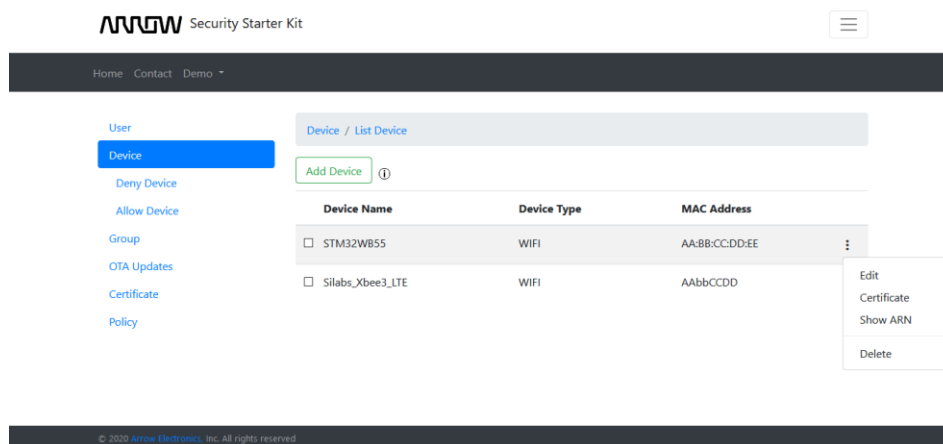


Figure 2: Listing devices

3.4 Deny Device

A Device which doesn't have at least one active certificate is added in the Deny Device list by MAC Address/UUID.

1. Deny by Certificate

You can remove a device from Deny list by attaching a certificate to it.

2. Deny by MAC Address

You can remove a device from Deny list by clicking the remove button.

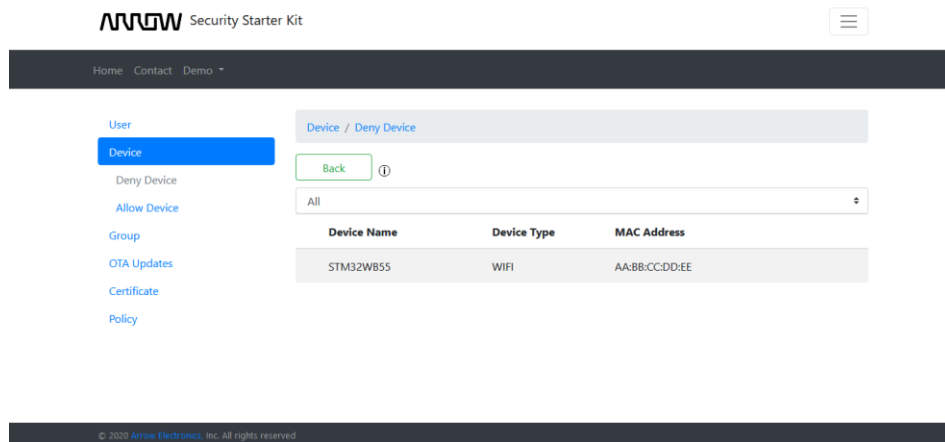


Figure 3: Listing blacklist devices

3.5 Allow Device

An Allowed Device is one which has at least one active certificate and is not added in the Deny Device list by MAC Address/UUID.

1. **Deny by Certificate**
You can add device to deny list by clicking 'Deny By Certificate.'
2. **Deny by MAC Address**
You can add device to deny list by clicking 'Deny By MAC Address/UUID'.

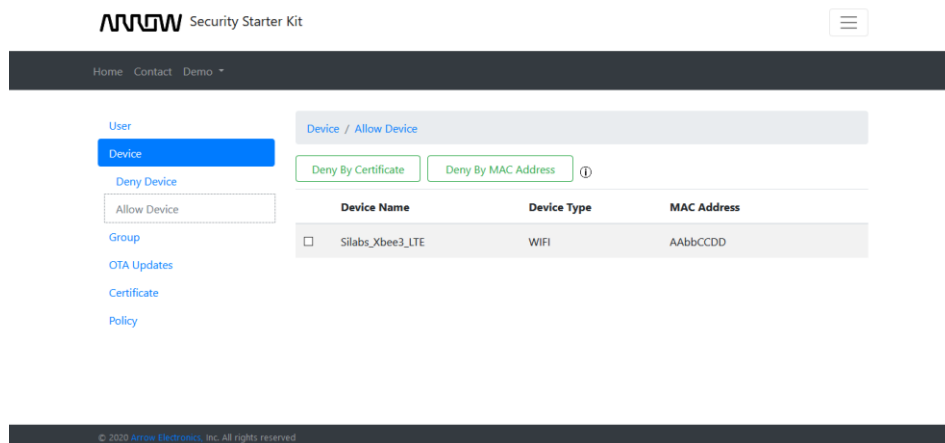


Figure 4: Listing whitelist devices

4 GREENGRASS

4.1 Description

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

4.2 Creating a Green grass Group

Setting up your Group requires you to provision a Core device in the IoT Registry, acquire a certificate for your Core, and assign an IAM role to your Group. A Group can be described by the following:

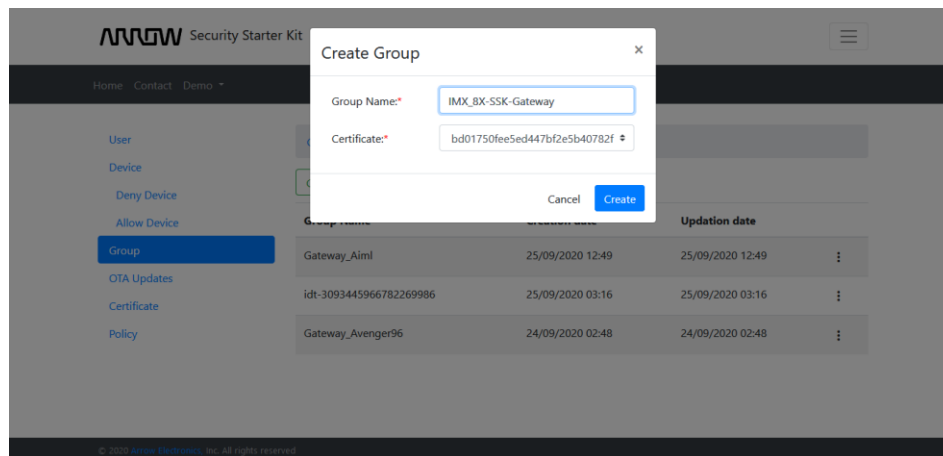


Figure 1: Create a Green grass Group

1. **Group Name**
The name of the Green grass Group
2. **Certificate Id**
The Certificate ID of the Green grass Core.

4.3 Listing Green grass Group

Listing of Greengrass group can be seen in following figure.

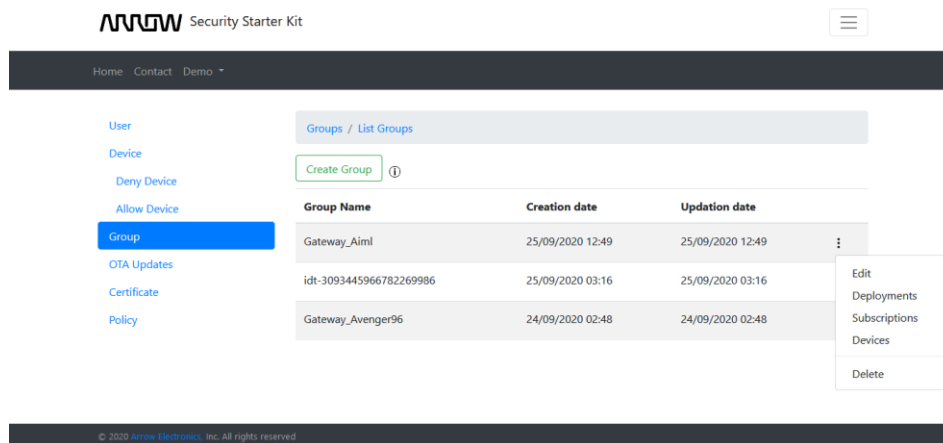


Figure 2: Listing Green grass Groups

4.4 Green grass Subscription

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

4.4.1 Creating a Subscription

A Subscription can be added to the group by the following figure:

1. **Source**
The name of the Source. Source can be Services like IoT Cloud, Local Shadow Service or Green grass devices.
2. **Topic**
The name of the Topic. AWS Cloud and a device can communicate on a given topic over MQTT.
3. **Target**
The name of the Target. Target can be Services like IoT Cloud, Local Shadow Service or Green grass devices.

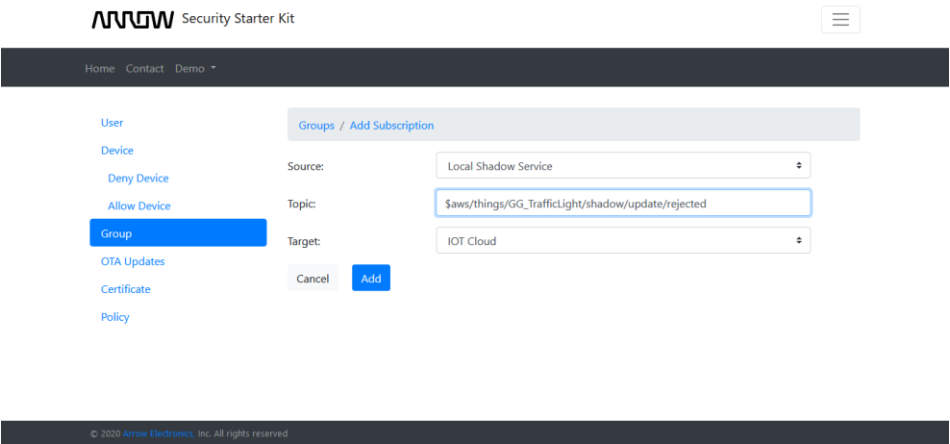


Figure 3: Create subscription To Group

4.4.2 Listing Subscriptions

A Subscription List can be seen in following figure with the path SSK -> Group:

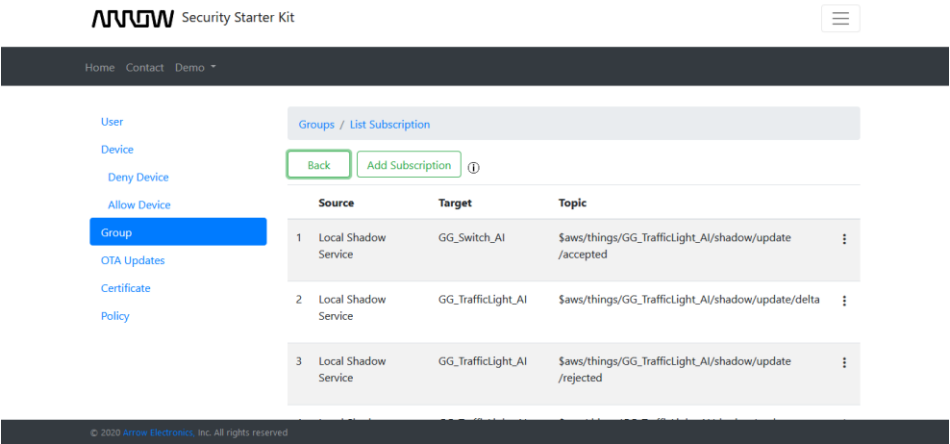


Figure 4: Listing subscriptions

4.5 Green grass Device

Greengrass Devices can be created by re-purposing an existing IoT Thing from your Registry or by creating new Registry items, and then adding them to a Greengrass Group.

4.5.1 Add a device

A device can be added to a group ~~by~~ as shown in the following figure:

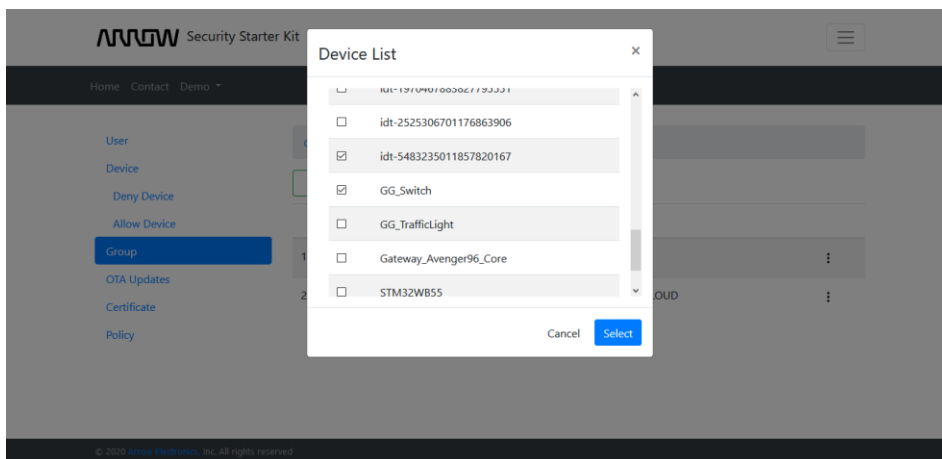


Figure 5: Add a device To Group

1. Device Name

The name of the device.

4.5.2 Listing Devices

A Device List can be seen in following figure with the path SSK -> Group:

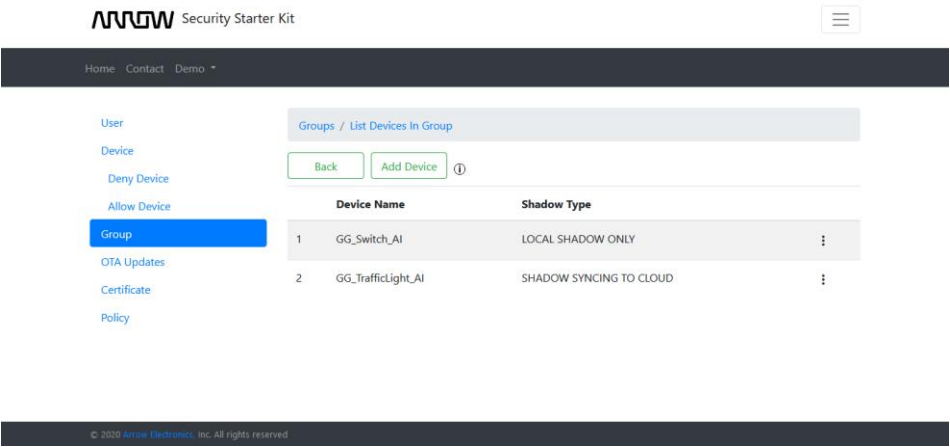


Figure 6: Listing devices

4.6 Green grass Deployment

A deployment of green grass group & core to a device can be ~~setup by~~seen in the following figure:

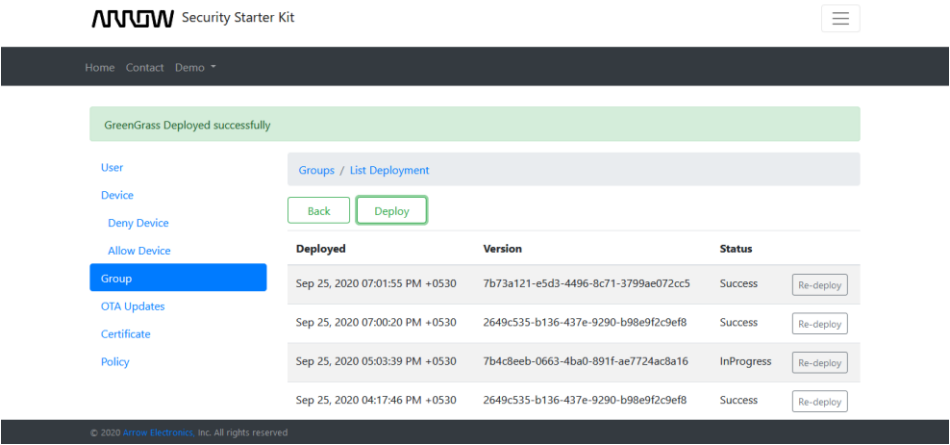


Figure 7: Deployment Greengrass

5 OTA UPDATES

5.1 Description

AWS IoT Device Management job orchestration and notification service allows you to define a set of remote operations called jobs that are sent to and executed on one or more devices connected to AWS IoT.

5.2 Create a FreeRTOS OTA update job (Schedule OTA)

This Over-the-air (OTA) update job will send your firmware image securely over MQTT or HTTP to FreeRTOS-based devices.

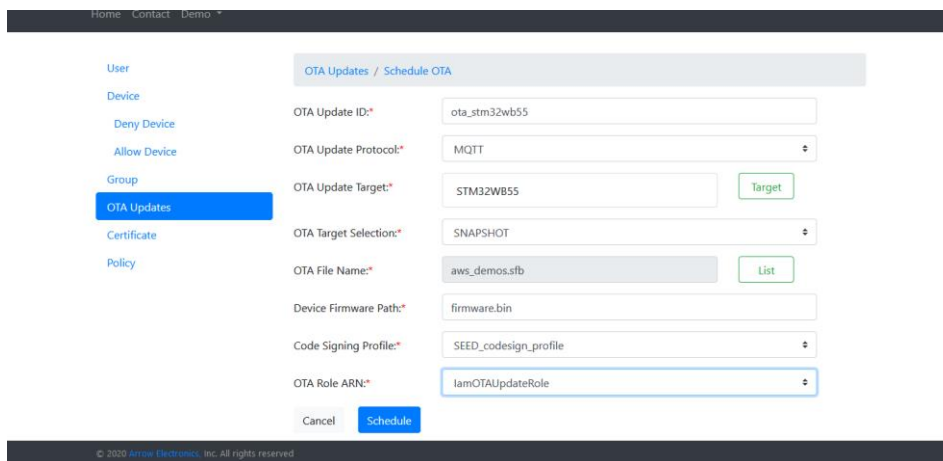


Figure 1: Schedule OTA

1. **OTA Update ID**
A unique OTA update ID.
2. **OTA Update Protocol**
The protocol that you choose must be supported by your device. If you select a protocol that is not supported by your device, the firmware update will be unsuccessful.
3. **OTA Update Target**
Select the devices you want to include in this job.
4. **OTA Target Selection**
Snapshot job is sent to all targets that were selected when you created the job. After those targets complete the job (or report that they are unable to do so), the job is complete.

5. OTA File Name

Name of firmware.

6. Device Firmware Path

This is the location and name to use when storing the firmware on the FreeRTOS device during OTA update. It is an optional field since certain devices do not store the image to a filesystem and may instead write directly to internal flash memory.

7. Code Signing Profile

Code signing ensures that devices only run code published by trusted authors and that the code has not been altered or corrupted since it was signed. You have three options for code signing.

8. OTA Role ARN

A Role which grants AWS IoT access to the S3, AWS IoT jobs and AWS Code signing resources to create an OTA update job.

5.3 Create Custom Job (Schedule 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.

The screenshot shows the 'Arrow Security Starter Kit' web application. The top navigation bar includes 'Home', 'Contact', and 'Demo'. The left sidebar lists navigation options: 'User', 'Device', 'Deny Device', 'Allow Device', 'Group', 'OTA Updates' (highlighted), 'Certificate', and 'Policy'. The main content area is titled 'OTA / Schedule Custom Job' and contains the following form fields:

- Job Create ID:** A text input field containing 'P2P'.
- Job Target:** A text input field containing 'STM32WB55', with a green 'Target' button next to it.
- Job Document Name:** A text input field containing 'aws_ota_v1.1.gbl', with a green 'List' button next to it.
- Job Target Selection:** A dropdown menu showing 'SNAPSHOT'.

At the bottom of the form are two buttons: 'Cancel' and 'Schedule Job'.

© 2020 Arrow Electronics, Inc. All rights reserved.

Figure 2: Schedule custom job

1. Job Create ID

A unique job Id.

2. Job Target

Select the devices you want to include in this job.

3. Job Document Name

Upload a job file that defines what your job should do.

Job documents are JSON documents and should contain any information your devices need to perform a job. For example, a job document can contain one or more URLs where the device can download an update or some other data.

4. Job Target Selection

Snapshot job is sent to all targets that were selected when you created the job. After those targets complete the job (or report that they are unable to do so), the job is complete.

5.4 Listing OTA UPDATES

Listing of OTA jobs can be seen in following figure with the path SSK -> OTA Updates:

The screenshot shows the 'Schedule OTA' form in the SSK interface. The left sidebar contains a navigation menu with the following items: User, Device, Deny Device, Allow Device, Group, OTA Updates (highlighted in blue), Certificate, and Policy. The main content area is titled 'OTA Updates / Schedule OTA' and contains the following fields:

- OTA Update ID: ota_stm32wb55
- OTA Update Protocol: MQTT
- OTA Update Target: STM32WB55 (with a 'Target' button)
- OTA Target Selection: SNAPSHOT
- OTA File Name: aws_demos.sfb (with a 'List' button)
- Device Firmware Path: firmware.bin
- Code Signing Profile: SEED_codesign_profile
- OTA Role ARN: iamOTAUpdateRole

At the bottom of the form are 'Cancel' and 'Schedule' buttons. The footer of the page reads '© 2020 Arrow Electronics, Inc. All rights reserved'.

Figure 3: Listing OTA Job

5.5 Create a Code Signing profile

The code signing profile contains information needed to create a code signing job. It specifies your device's hardware platform, certificate from AWS Certificate Manager, and the location of your code signing certificate path on your device.

The screenshot shows the 'Create Code Signing Profile' dialog box. The fields are filled with the following values:

- Profile Name: SEED_codesign_profile
- Platform ID: Amazon FreeRTOS SHA256-ECDSA
- Code Signing Certificate: ecdsasigner.crt
- Certificate Private Key: ecdsasigner.key
- Path Name: /home/certificate/ecdsasigner.crt

The background interface shows a sidebar with options like User, Device, Deny Device, Allow Device, Group, OTA Updates, Certificate, and Policy. A table of existing profiles is visible below the dialog:

ID	Profile Name	Platform ID	Status
4	VireshCodeSign2	AmazonFreeRTOS-Default	Active
5	wb55_16july_profile	AmazonFreeRTOS-Default	Active

Figure 4: Create code signing profile

- Profile Name**
A unique profile name.
- Platform ID**
Select the platform.
- Code Signing Certificate**
Upload signing certificate
- Certificate Private Key**
Upload private key
- Path Name**
Select the path name to the certificate on the device

6 CERTIFICATE

6.1 Description

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

6.2 Register CA

To use your own X.509 certificates, you must register a CA certificate with AWS IoT. The CA certificate can then be used to sign device certificates. You can register up to ten CA certificates with the same subject field and public key per AWS account. This allows you to have more than one CA sign your device certificates.

- Step 1: Generate a key pair for the private key verification certificate
- Step 2: Copy this registration code (From Setting page at top-right corner)
- Step 3: Create a CSR with this registration code. Put the registration code in the Common Name field
- Step 4: Use the CSR that was signed with the CA private key to create a private key verification certificate
- Step 5: Upload the CA certificate (rootCA.pem)
- Step 6: Upload the verification certificate (verificationCert.crt) :

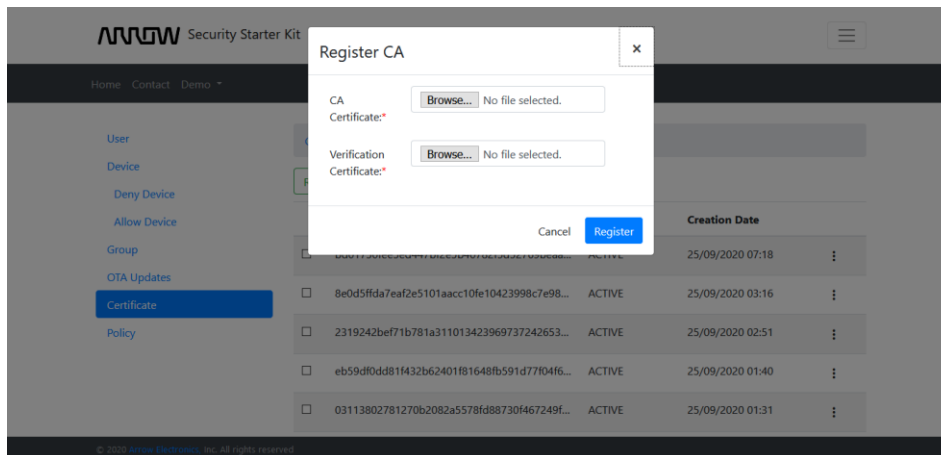


Figure 1: Register CA

1. **Root CA File**
Select & upload root CA file.
2. **Verification Certificate File**
Select & upload verification certificate file.

6.3 Listing Certificates

Listing of all registered certificates can be seen in following figure with the path SSK -> Certificate:

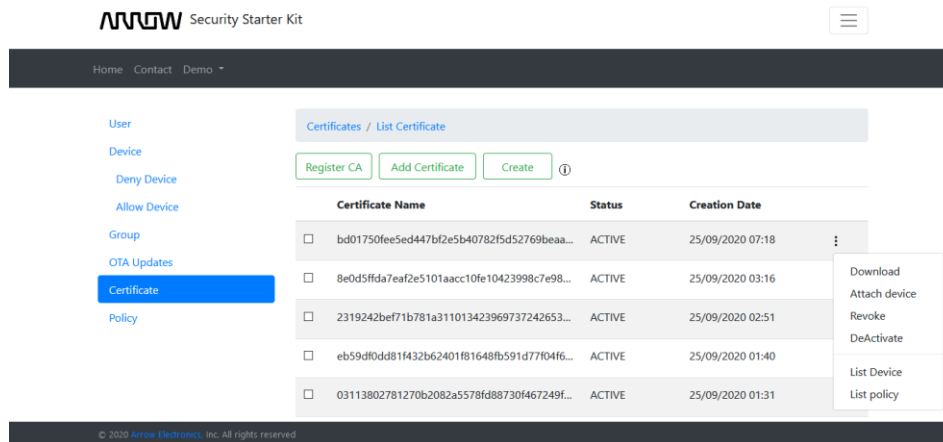


Figure 2: Listing Certificates

Certificate will be having following operations: On the Certificate page, a user can perform the following operations:

1. Download
Download certificate file in PEM format.

2. Attach device

Attach certificate to device.

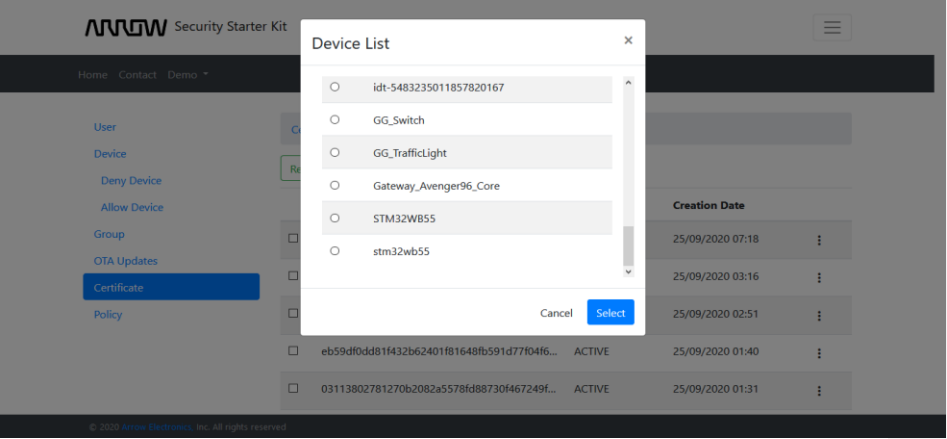


Figure 3: Attach certificate to Device

3. Revoke

Revokes a specific certificate. Once a certificate is revoked it cannot be reactivated, it can only be deleted.

Formatted: Font color: Accent 5

Formatted: Font color: Auto, Highlight

Formatted: Font color: Auto, Highlight

4. Deactivate

Changes status to inactive. If the certificate has been revoked it cannot be made inactive.

5. Delete

Deletes a certificate. You cannot delete an active certificate.

Formatted: Font color: Auto

6. List Device

Lists all devices attached to a certificate.

7. List Policy

Lists all policies attached to a certificate.

6.4 Add a Certificate

Select or register the CA certificate used to sign your device certificates. To use device certificates that are not signed by a registered CA, just select next.

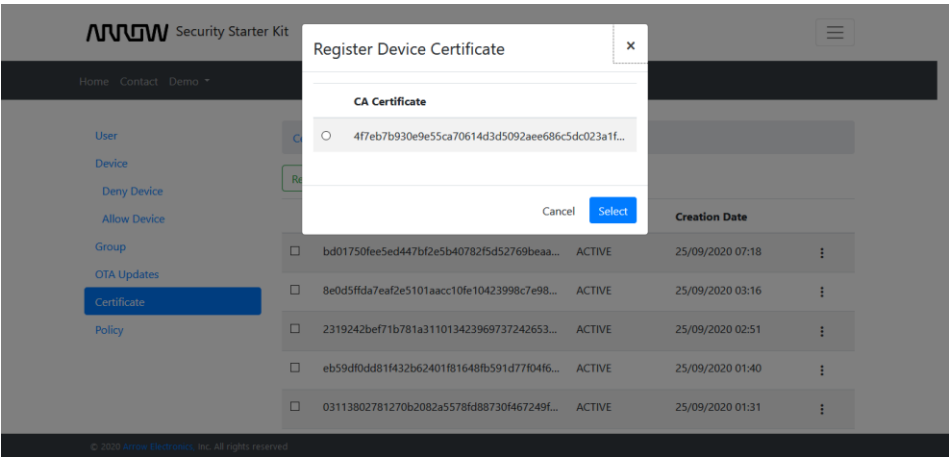


Figure 4: Add Certificate Part 1

Paste the existing certificate PEM file.

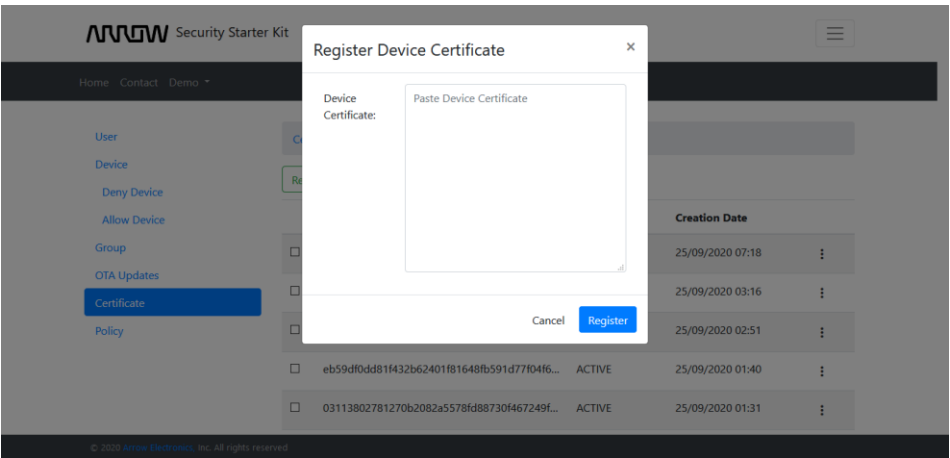


Figure 5: Add Certificate part 2

1. Registered CA
Select CA ID from list of CA.
2. Existing Certificate File

Paste the Certificate PEM file.

6.5 Create a Certificate (One Click)

This will generate a certificate, public key, and private key using AWS IoT's certificate authority.

7 POLICY

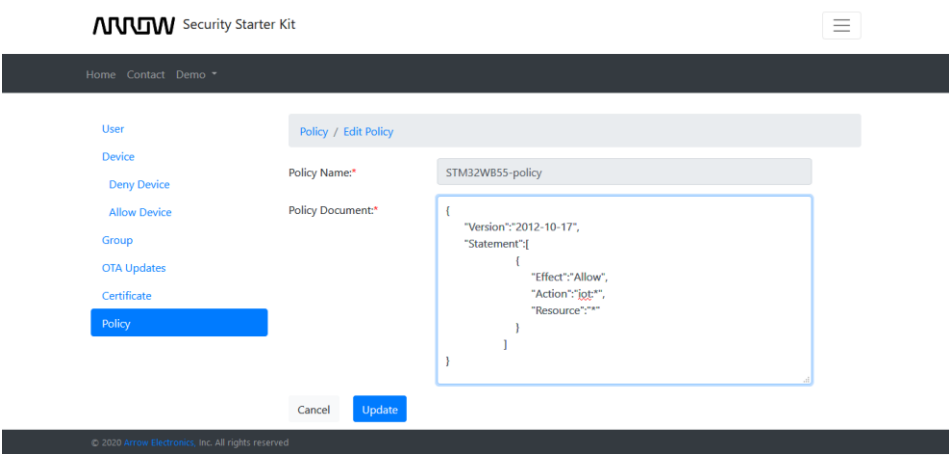
7.1 Description

AWS IoT policies grant or deny access to AWS IoT resources such as things, thing shadows, and MQTT topics. A device or user can invoke AWS IoT operations only if they are granted the appropriate permissions.

Policies give permissions to AWS IoT clients regardless of the authentication mechanism they use to connect to AWS IoT. To control which resources a device can access, attach one or more AWS IoT policies to the certificate associated with the device.

7.2 Creating a Policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters).



The screenshot shows the 'Create Policy' interface of the AWS IoT Security Starter Kit. On the left is a navigation menu with links for User, Device, Deny Device, Allow Device, Group, OTA Updates, Certificate, and Policy (which is highlighted in blue). The main area has a breadcrumb 'Policy / Edit Policy' and two input fields: 'Policy Name' with the value 'STM32WB55-policy' and 'Policy Document' containing a JSON policy. The JSON policy is:

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

 At the bottom are 'Cancel' and 'Update' buttons. The footer shows '© 2020 Arrow Electronics, Inc. All rights reserved.'

Figure 1: Create Policy

1. **Policy Name**
The name of the policy.
2. **Policy Document**
Policy document defines the types of actions that can be performed by a resource. It will not allow any spaces

Policy document Description can be seen below:

Policy document has three main elements

i. Effect – Allow/ Deny

ii. Action

Action can be defined by following types:

- a. iot:*
- b. iot:Publish
- c. iot:Subscribe
- d. iot:Connect
- e. iot:Receive
- f. iot:UpdateThingShadow
- g. iot:GetThingShadow
- h. iot:DeleteThingShadow

iii. Resource ARN

Resource could be client ID ARN, topic ARN or topic filter ARN

7.3 Listing the Policy

Listing of policies can be seen in following figure with the path SSK -> Policy:

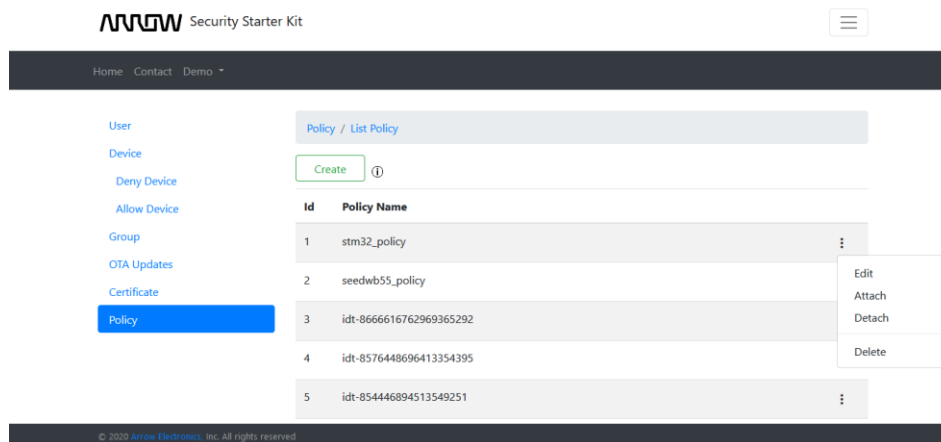


Figure 2: Listing Green grass Groups

7.4 Attach Policy

Policys can be attached to resources like Certificates, which can be ~~seen in~~seen in the following figure:

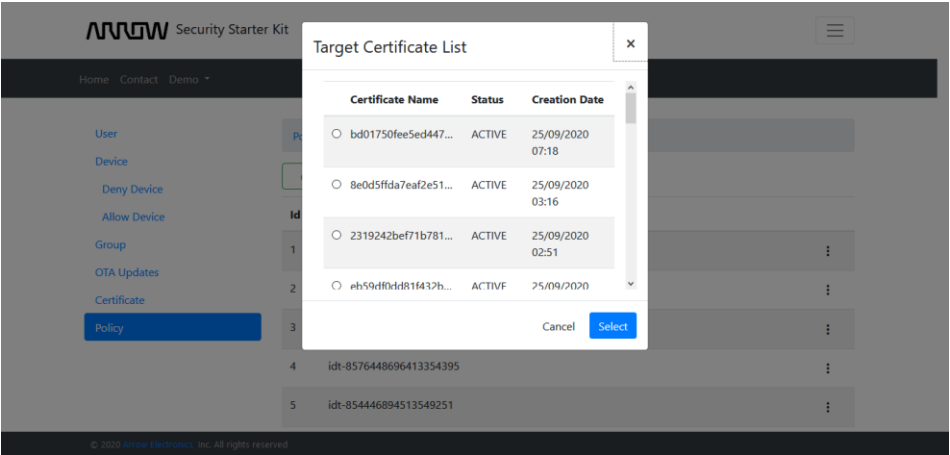


Figure 3: Attach policy to Certificate

7.5 Detach Policy

Detach policys from a resource like Certificates, which can be seen here in the following figure:

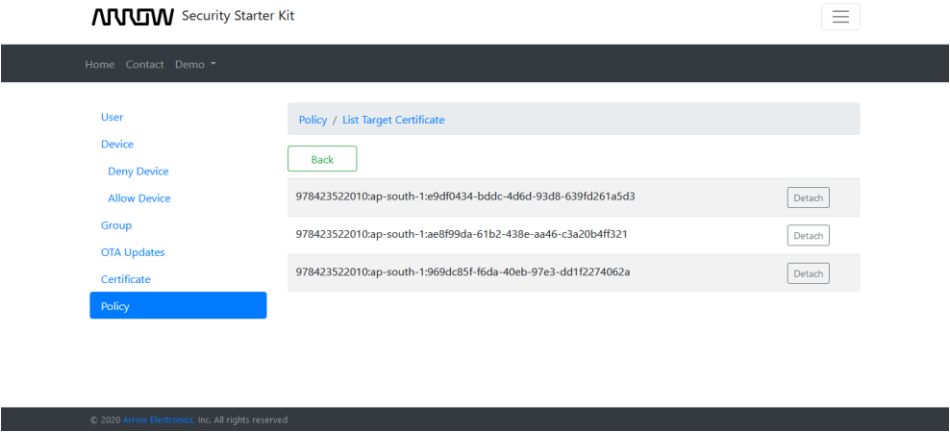


Figure 4: Detach policy from resource

8 REFERENCES

- [1] <https://docs.aws.amazon.com/greengrass/latest/developerguide/gg-dg.pdf>
- [2] <https://aws.amazon.com/blogs/iot/using-a-trusted-platform-module-for-endpoint-device-security-in-aws-iot-greengrass/>
- [3] <https://docs.aws.amazon.com/iot/latest/developerguide/register-CA-cert.html>
- [4] <https://aws.amazon.com/console/>