AWSOTAUG

Amazon FreeRTOS Over-The-Air Updates using i.MX RT1060

Rev. 4 — 10 January 2024 User guide

Document information

Information	Content
Keywords	OTAP, AWS, AWS services, Amazon FreeRTOS, Over The Air, RT1060-EVK SDK
Abstract	This document lists the steps to configure AWS services to make an Amazon FreeRTOS Over The Air Update using NXPs RT1060-EVK SDK

1 Overview

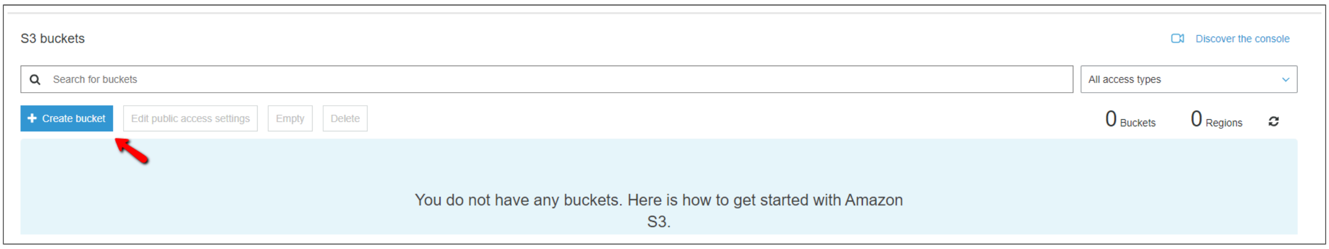
This guide walks through the steps to configure AWS services to make an Amazon FreeRTOS Over The Air Update using NXPs RT1060-EVK SDK. First, it creates an IAM role with OTA update, S3, IoT policies, and permissions. Then, using OpenSSL and AWS CLI commands, a code signing certificate is issued. Finally, it shows how to create an IoT thing with the code signing certificate with an OTA job.

Note: *The figures used in the document might be slightly different due to background changes by Amazon.*

2 AWS OTA prerequisites

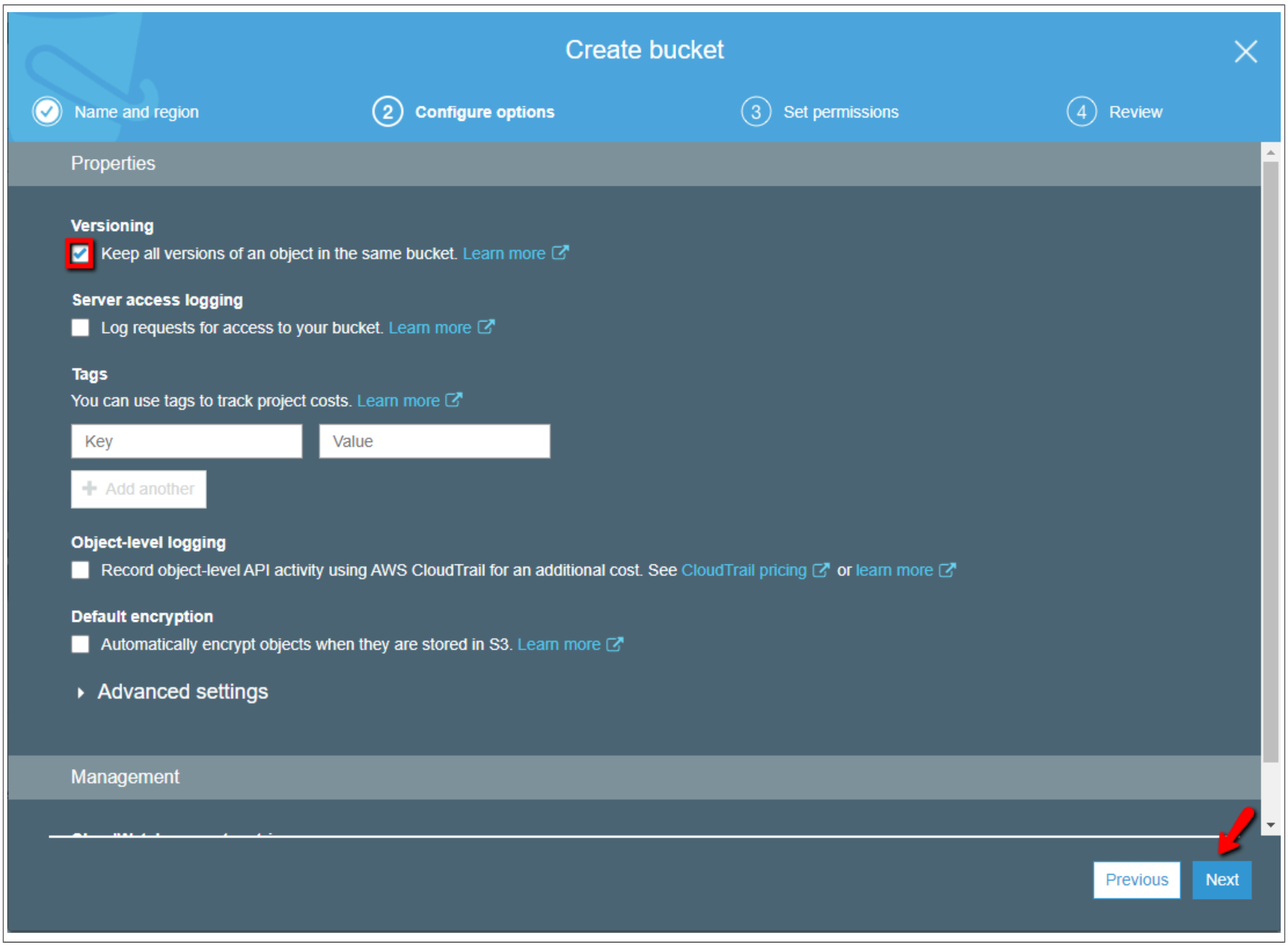
2.1 Create an Amazon S3 bucket and store your update

1. Go to the https://console.aws.amazon.com/s3/.
2. Choose **Create bucket**.



You do not have any buckets. Here is how to get started with Amazon S3.

3. Type a bucket name, and then choose **Next**.
4. Select **Versioning** to keep all versions in the same bucket, and then choose **Next**.



Versioning
☑ Keep all versions of an object in the same bucket. Learn more

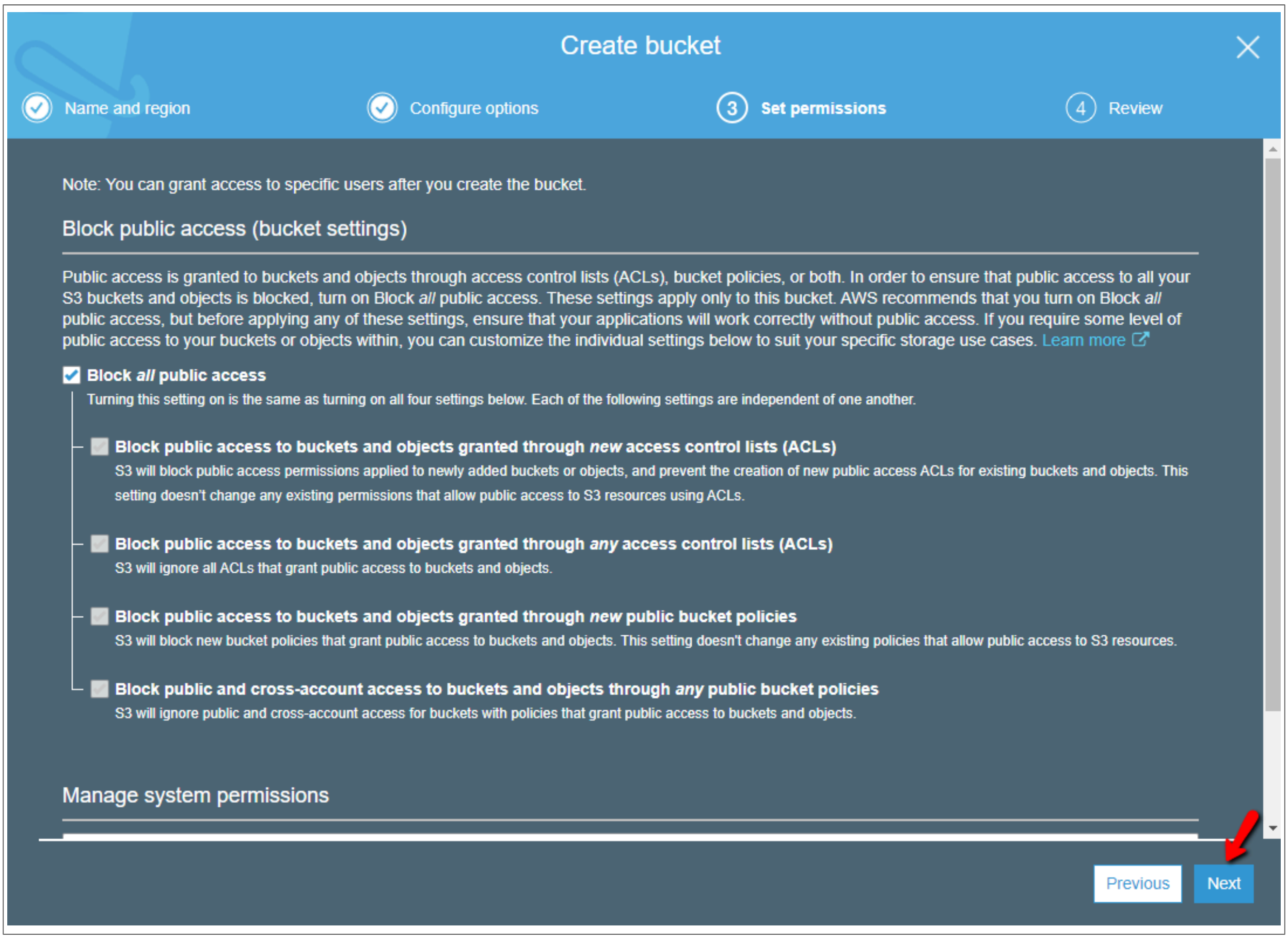
Server access logging
□ Log requests for access to your bucket. Learn more

Tags
You can use tags to track project costs. Learn more

Object-level logging
□ Record object-level API activity using AWS CloudTrail for an additional cost. See CloudTrail pricing or learn more

Default encryption
□ Automatically encrypt objects when they are stored in S3. Learn more

5. Choose **Next** to accept the default permissions.



Note: You can grant access to specific users after you create the bucket.

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, or both. 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. 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

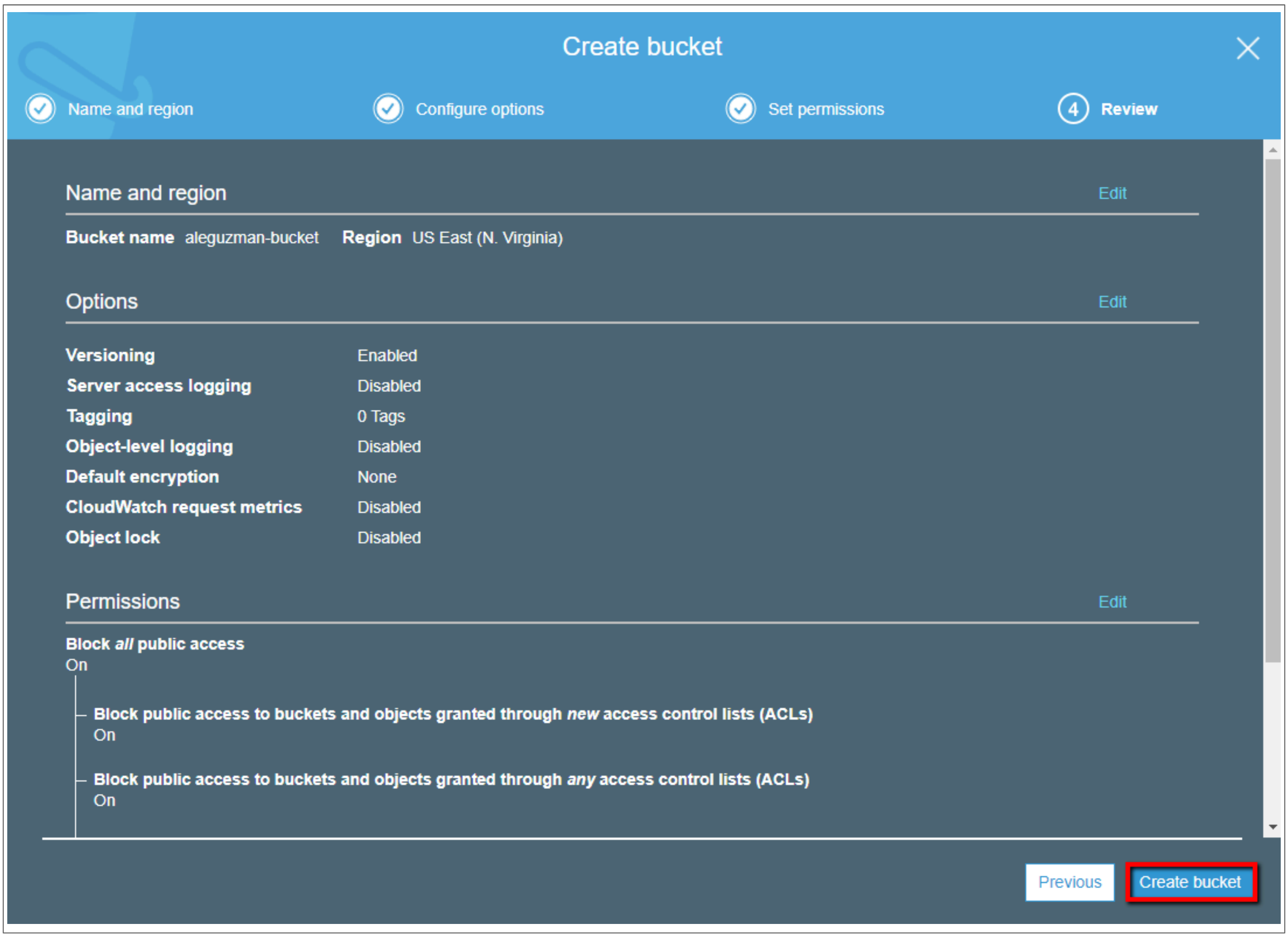
☑ **Block *all* public access**

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 policies**
  S3 will block new bucket 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 policies**
  S3 will ignore public and cross-account access for buckets with policies that grant public access to buckets and objects.

Manage system permissions

6. Choose **Create bucket**.



Name and region

Bucket name aleguzman-bucket **Region** US East (N. Virginia)

Options

Versioning	Enabled
Server access logging	Disabled
Tagging	0 Tags
Object-level logging	Disabled
Default encryption	None
CloudWatch request metrics	Disabled
Object lock	Disabled

Permissions

Block *all* public access
On

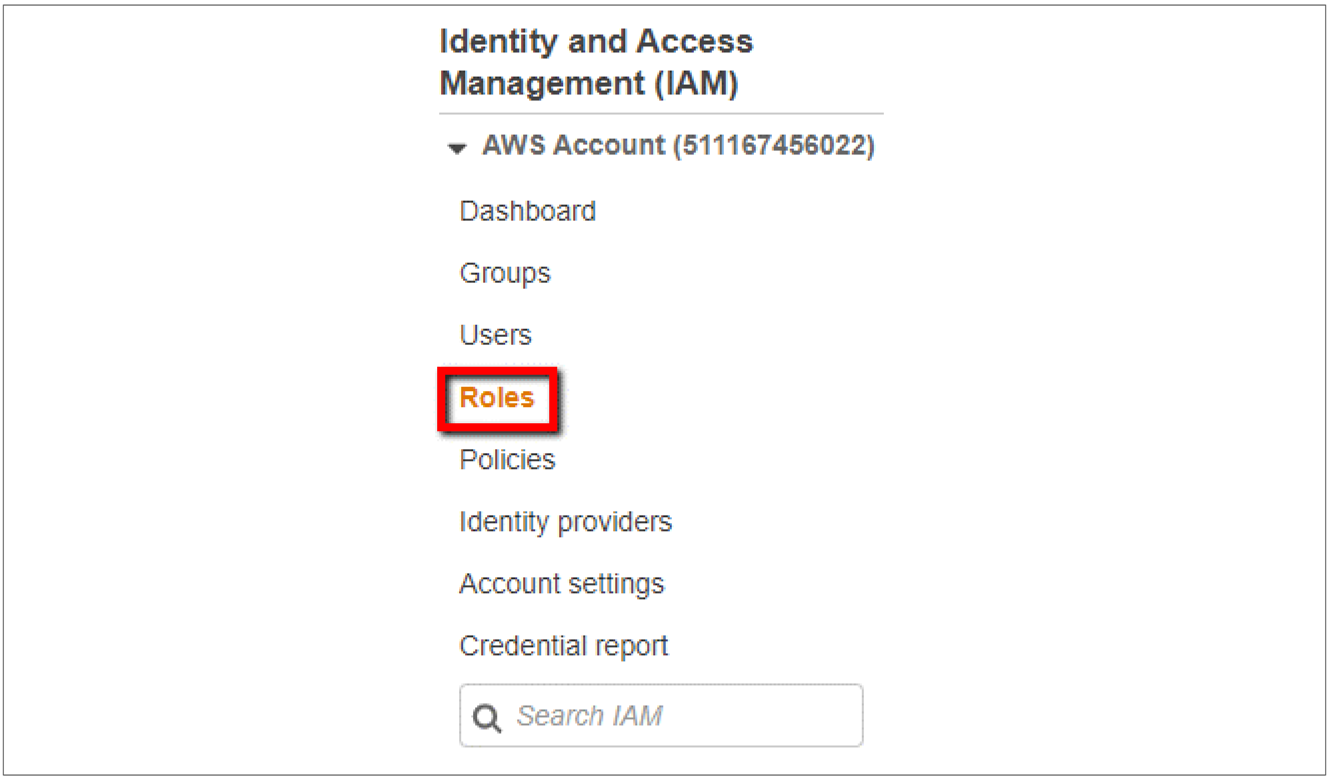
Block public access to buckets and objects granted through *new* access control lists (ACLs)
On

Block public access to buckets and objects granted through *any* access control lists (ACLs)
On

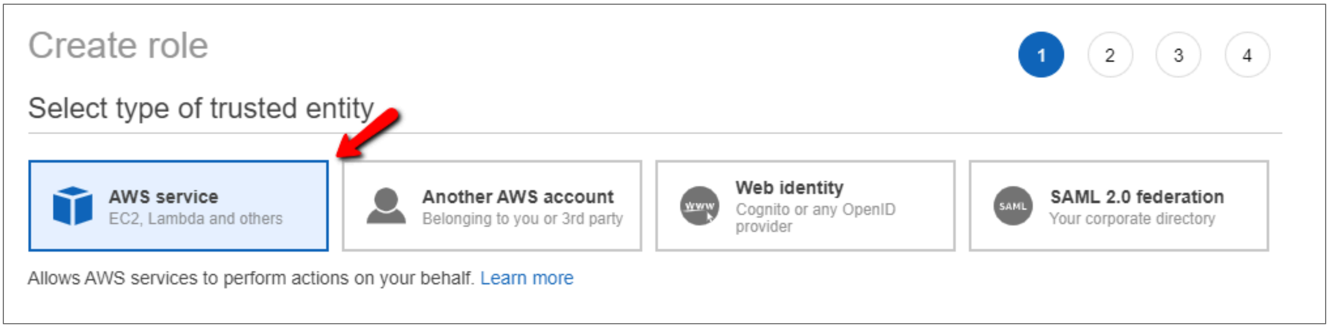
2.2 Create an OTA update service role

2.2.1 Create an OTA service role

1. Sign in to the https://console.aws.amazon.com/iam/.
2. From the navigation pane, choose **Roles**.

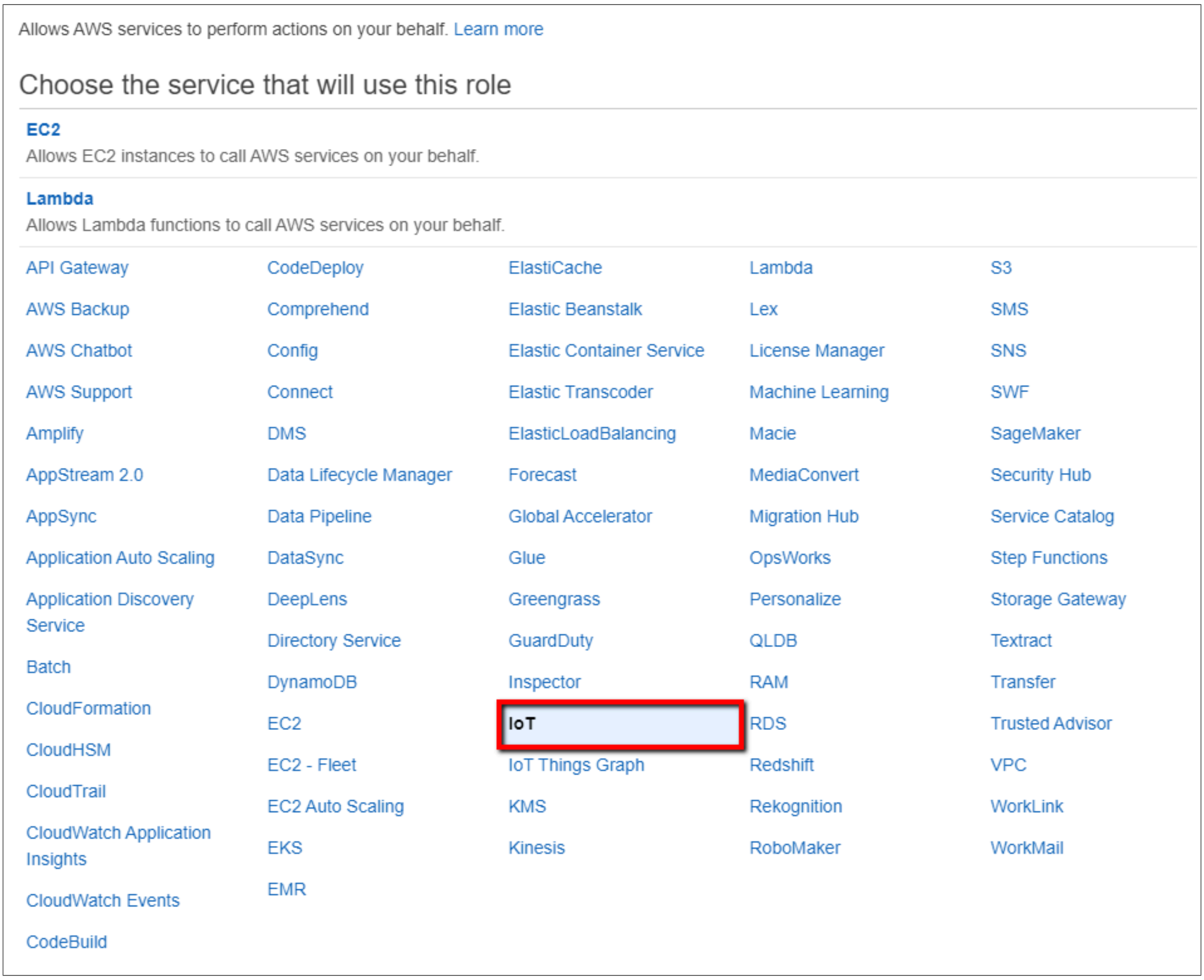


3. Choose to **Create role**.
4. Under **Select type of trusted entity**, choose **AWS Service**.



Allows AWS services to perform actions on your behalf. Learn more

5. Choose **IoT** from the list of AWS services.



Allows AWS services to perform actions on your behalf. Learn more

Choose the service that will use this role

EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

API Gateway
AWS Backup
AWS Chatbot
AWS Support
Amplify
AppStream 2.0
AppSync
Application Auto Scaling
Application Discovery Service
Batch
CloudFormation
CloudHSM
CloudTrail
CloudWatch Application Insights
CloudWatch Events
CodeBuild

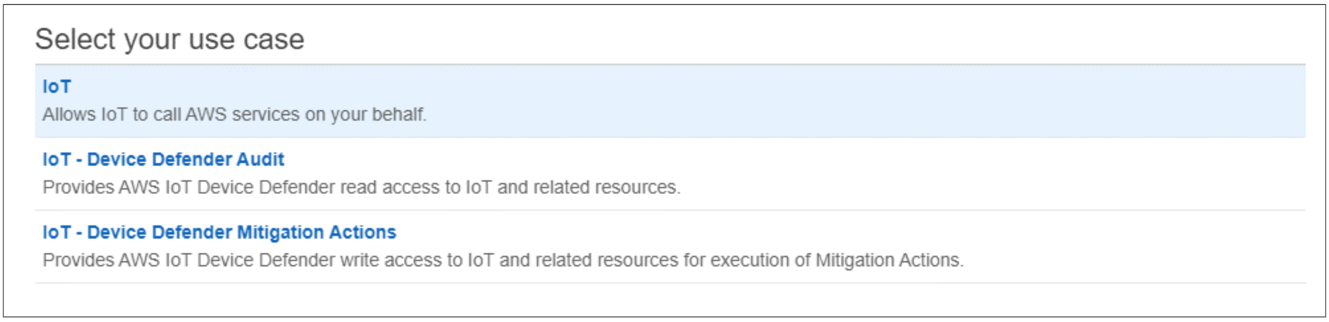
CodeDeploy
Comprehend
Config
Connect
DMS
Data Lifecycle Manager
Data Pipeline
DataSync
DeepLens
Directory Service
DynamoDB
EC2
EC2 - Fleet
EC2 Auto Scaling
EKS
EMR

ElastiCache
Elastic Beanstalk
Elastic Container Service
Elastic Transcoder
ElasticLoadBalancing
Forecast
Global Accelerator
Glue
Greengrass
GuardDuty
Inspector
IoT
IoT Things Graph
KMS
Kinesis

Lambda
Lex
License Manager
Machine Learning
Macie
MediaConvert
Migration Hub
OpsWorks
Personalize
QLDB
RAM
RDS
Redshift
Rekognition
RoboMaker

S3
SMS
SNS
SWF
SageMaker
Security Hub
Service Catalog
Step Functions
Storage Gateway
Textract
Transfer
Trusted Advisor
VPC
WorkLink
WorkMail

6. Under **Select your use case**, choose **IoT**.



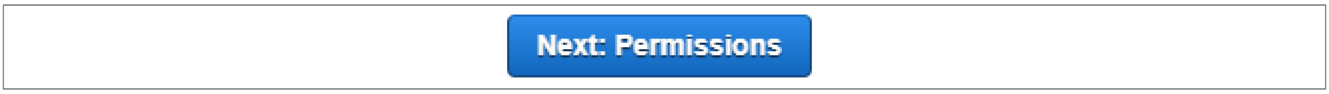
Select your use case

IoT
Allows IoT to call AWS services on your behalf.

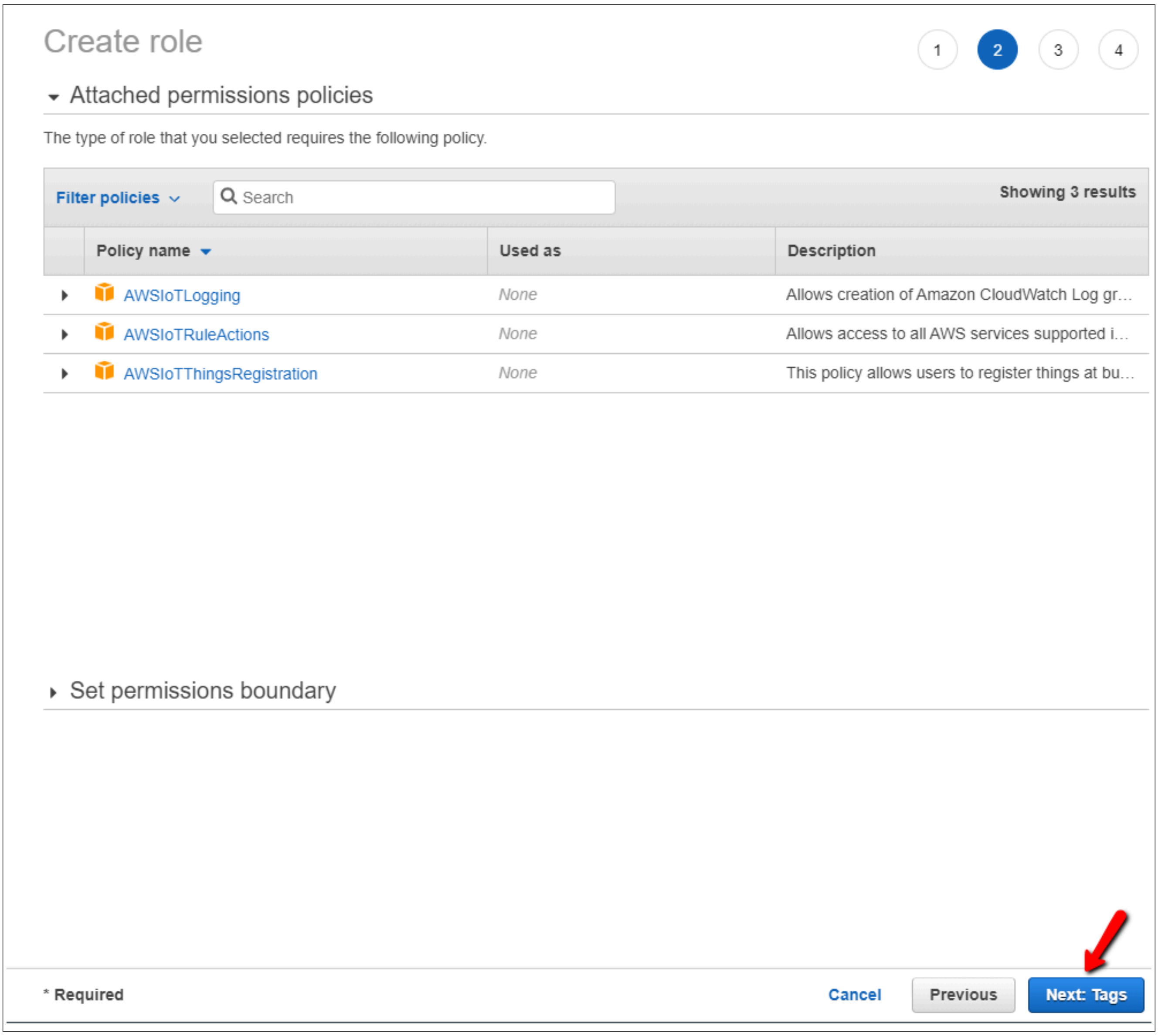
IoT - Device Defender Audit
Provides AWS IoT Device Defender read access to IoT and related resources.

IoT - Device Defender Mitigation Actions
Provides AWS IoT Device Defender write access to IoT and related resources for execution of Mitigation Actions.

7. Choose **Next: Permissions**.



8. Choose **Next: Tags**.



Create role

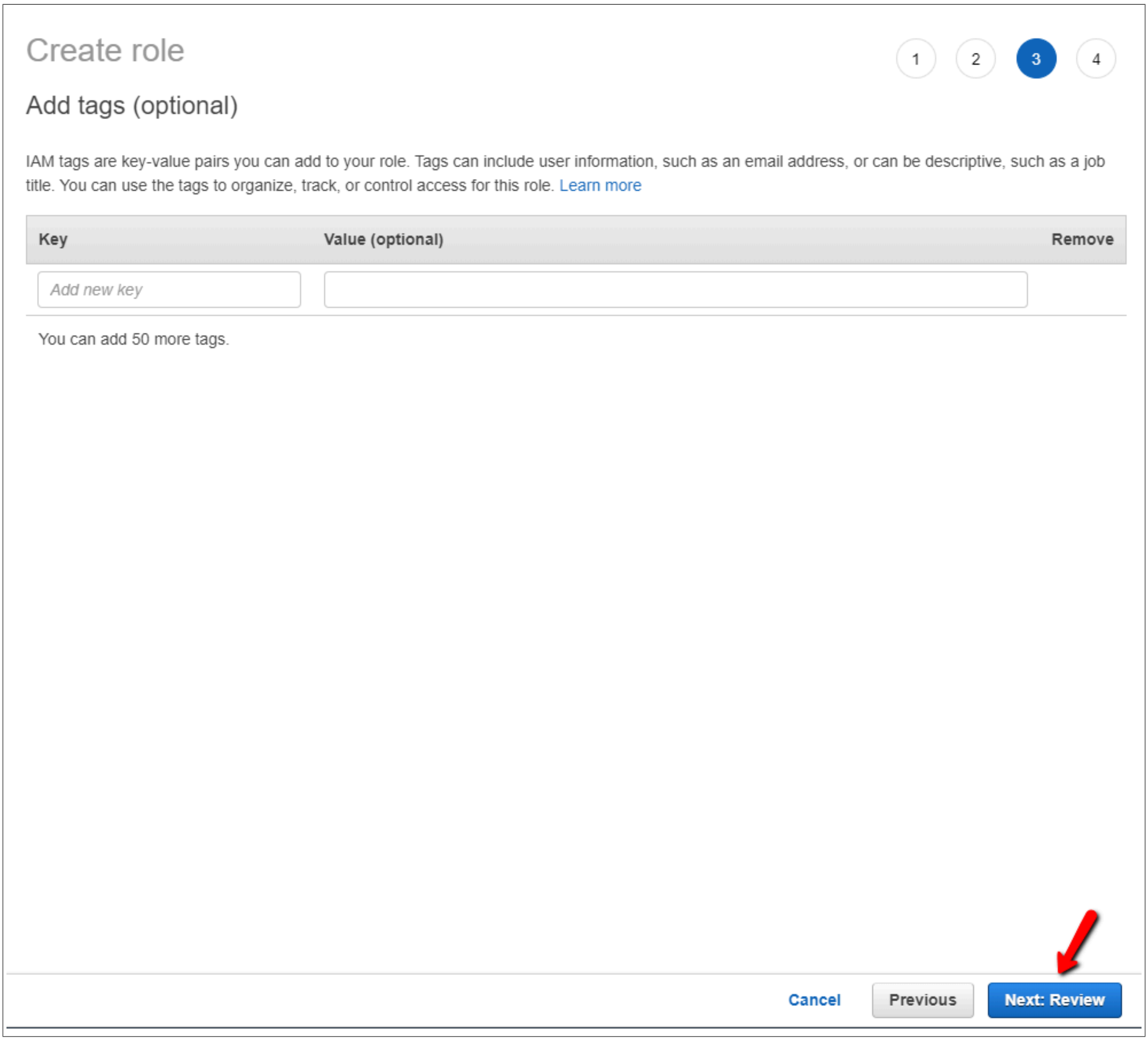
Attached permissions policies

The type of role that you selected requires the following policy.

Policy name	Used as	Description
AWSIoTLogging	None	Allows creation of Amazon CloudWatch Log gr...
AWSIoTRuleActions	None	Allows access to all AWS services supported i...
AWSIoTThingsRegistration	None	This policy allows users to register things at bu...

Set permissions boundary

9. Choose **Next: Review**.



Create role

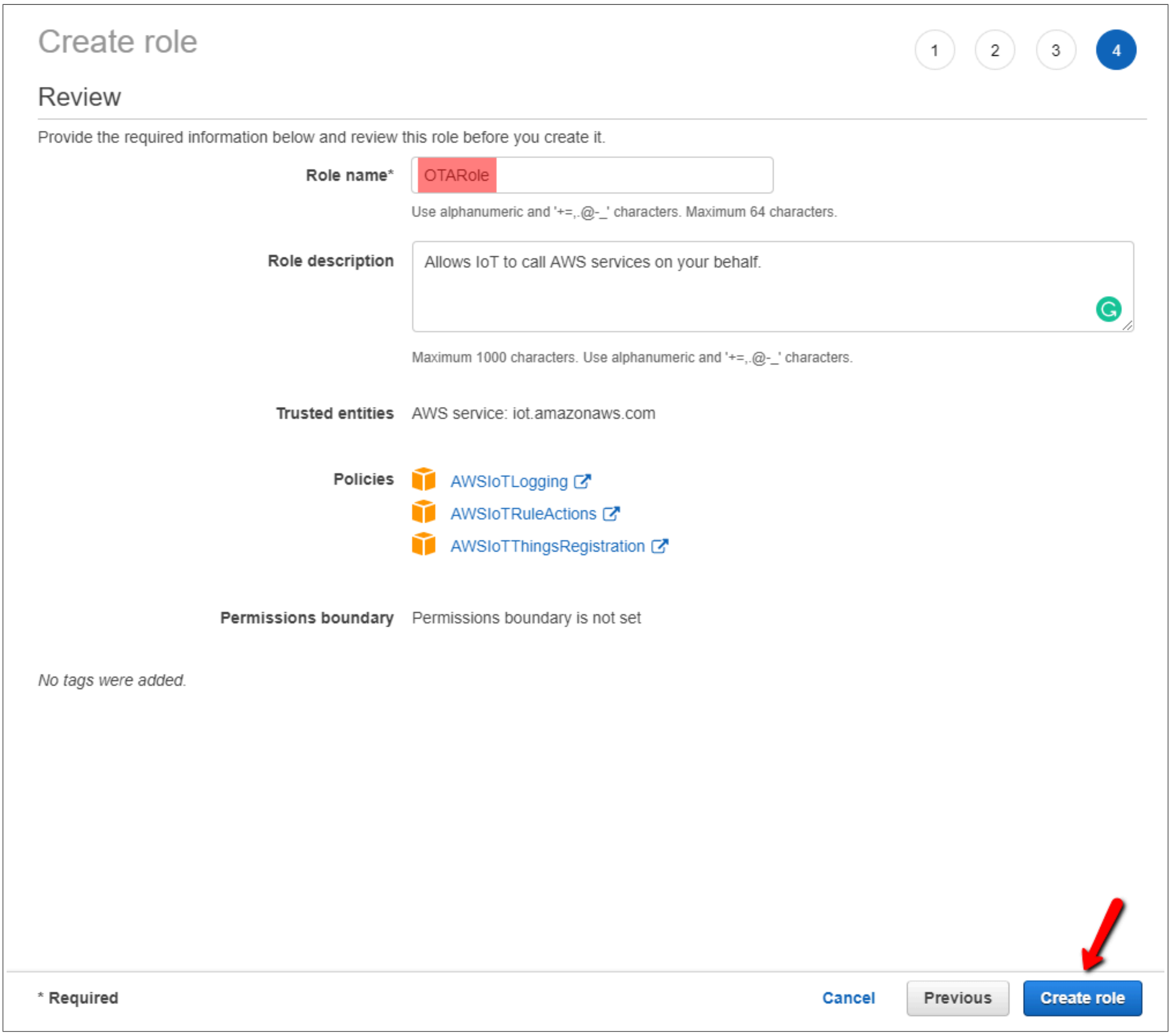
Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. Learn more

Key	Value (optional)	Remove
Add new key		

You can add 50 more tags.

10. Enter a role name and description and then choose to **Create role**.



Create role

Review

Provide the required information below and review this role before you create it.

Role name* OTARole

Use alphanumeric and '+=,.@-_' characters. Maximum 64 characters.

Role description Allows IoT to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Trusted entities AWS service: iot.amazonaws.com

Policies AWSIoTLogging
AWSIoTRuleActions
AWSIoTThingsRegistration

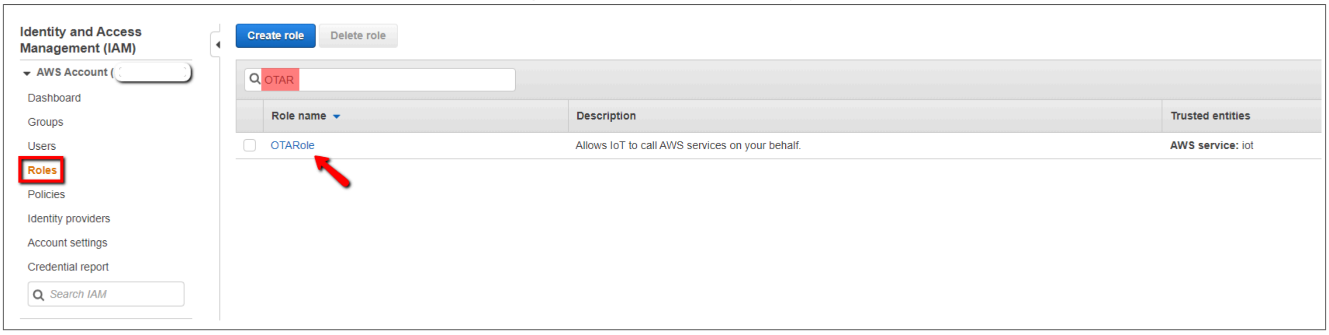
Permissions boundary Permissions boundary is not set

No tags were added.

* Required

2.2.2 To add OTA update permissions to your OTA service role

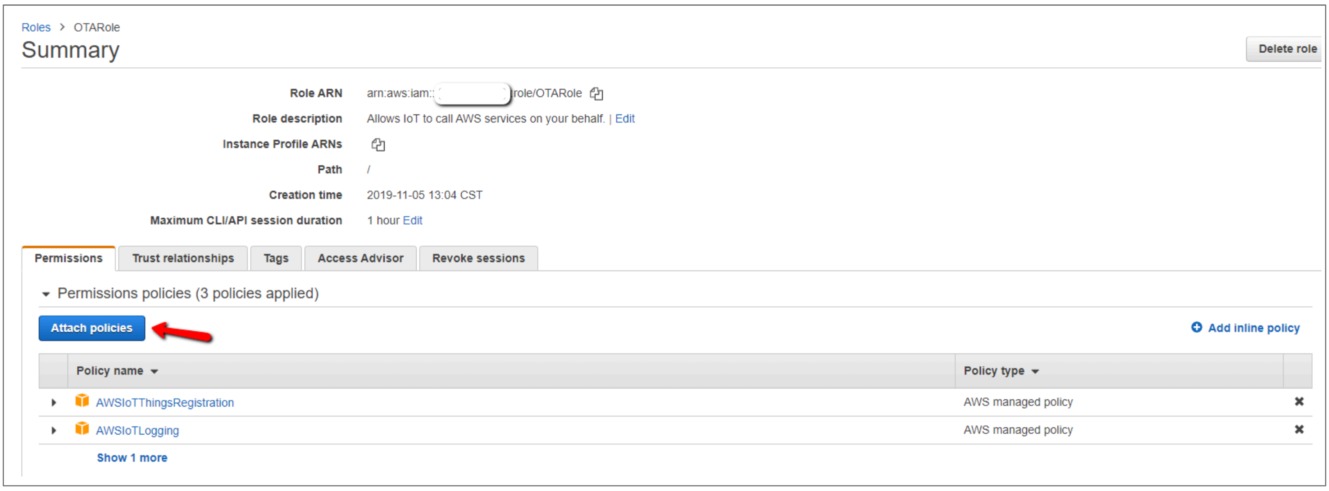
1. In the search box on the IAM console page, enter the name of your role, and then choose it from the list.



OTAR

Role name	Description	Trusted entities
OTARole	Allows IoT to call AWS services on your behalf.	**AWS service:** iot

2. Choose **Attach policies**.



Roles > OTARole

Summary

Role ARN: arn:aws:iam::[redacted]role/OTARole

Role description: Allows IoT to call AWS services on your behalf. | Edit

Instance Profile ARNs:

Path: /

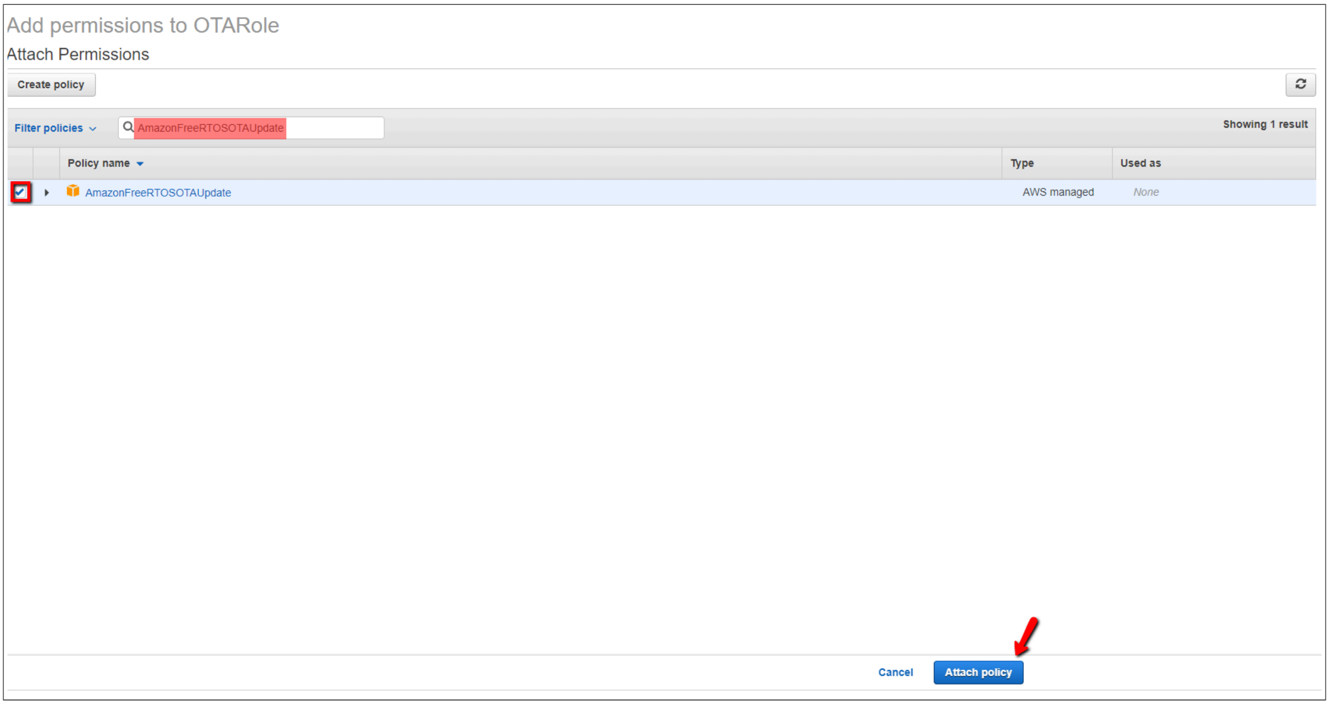
Creation time: 2019-11-05 13:04 CST

Maximum CLI/API session duration: 1 hour Edit

Permissions policies (3 policies applied)

Policy name	Policy type
AWSIoTThingsRegistration	AWS managed policy
AWSIoTLogging	AWS managed policy

3. In the **Search** box, enter **AmazonFreeRTOSOTAUpdate**, select **AmazonFreeRTOSOTAUpdate**.
4. From the list of filtered policies, and then choose **Attach policy** to attach the policy to your service role.



Add permissions to OTARole

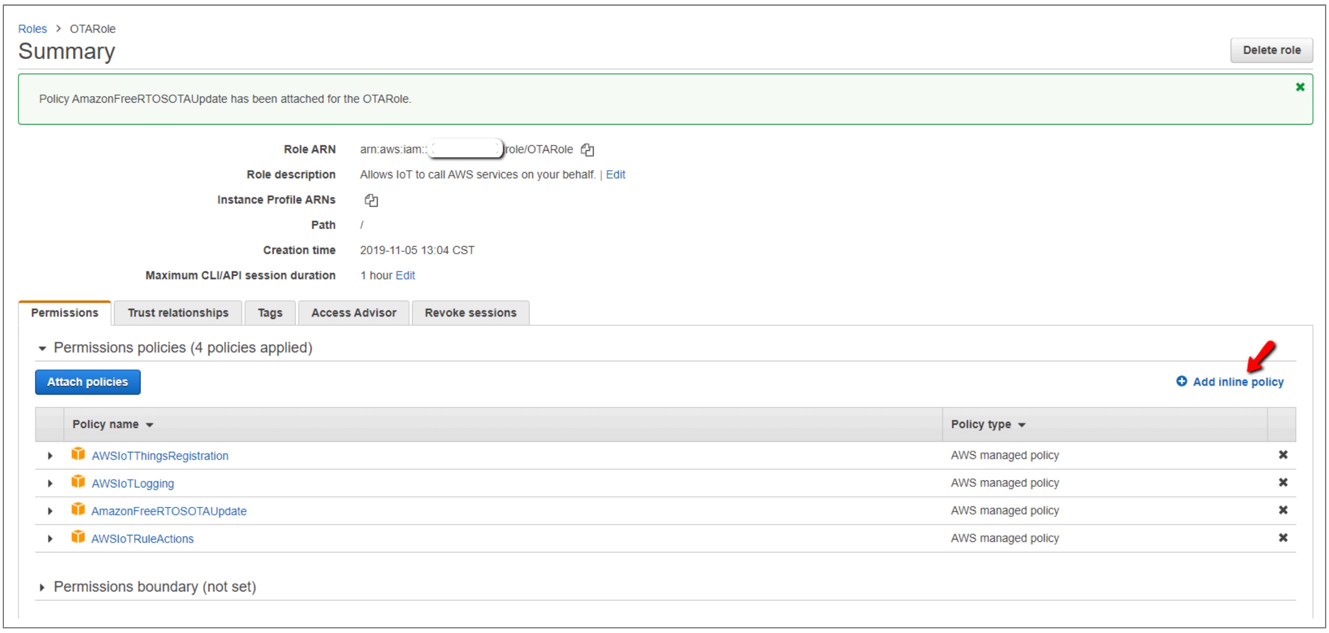
Attach Permissions

Filter policies: AmazonFreeRTOSOTAUpdate — Showing 1 result

	Policy name	Type	Used as
☑	AmazonFreeRTOSOTAUpdate	AWS managed	None

2.2.3 To add the required IAM permissions to your OTA service role

1. Choose **Add inline policy**.



Roles > OTARole

Summary

Delete role

Policy AmazonFreeRTOSOTAUpdate has been attached for the OTARole.

Role ARN	arn:aws:iam::[redacted]role/OTARole
Role description	Allows IoT to call AWS services on your behalf. | Edit
Instance Profile ARNs	
Path	/
Creation time	2019-11-05 13:04 CST
Maximum CLI/API session duration	1 hour Edit

Permissions policies (4 policies applied)

Policy name	Policy type
AWSIoTThingsRegistration	AWS managed policy
AWSIoTLogging	AWS managed policy
AmazonFreeRTOSOTAUpdate	AWS managed policy
AWSIoTRuleActions	AWS managed policy

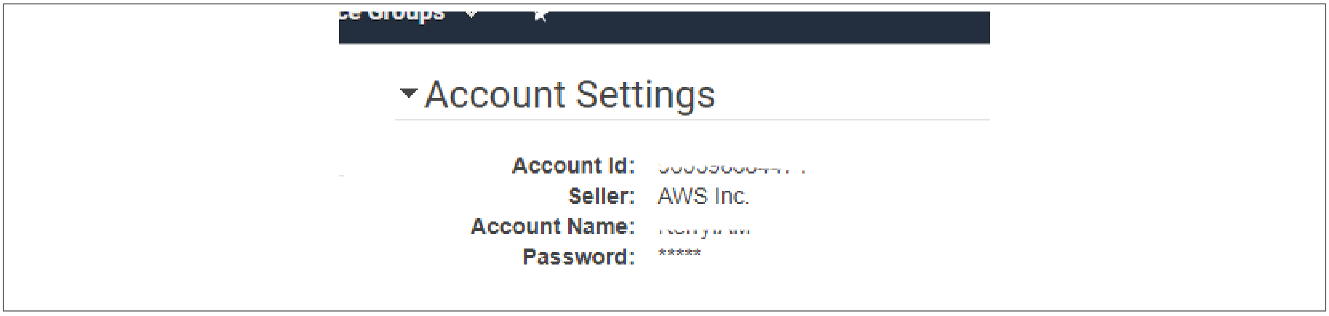
Permissions boundary (not set)

2. Choose the **JSON** tab.
3. Copy and paste the following policy document into the text box:

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

Make sure that you replace <*your_account_id*> with your AWS account ID, and <*your_role_name*> with the name of the OTA service role.

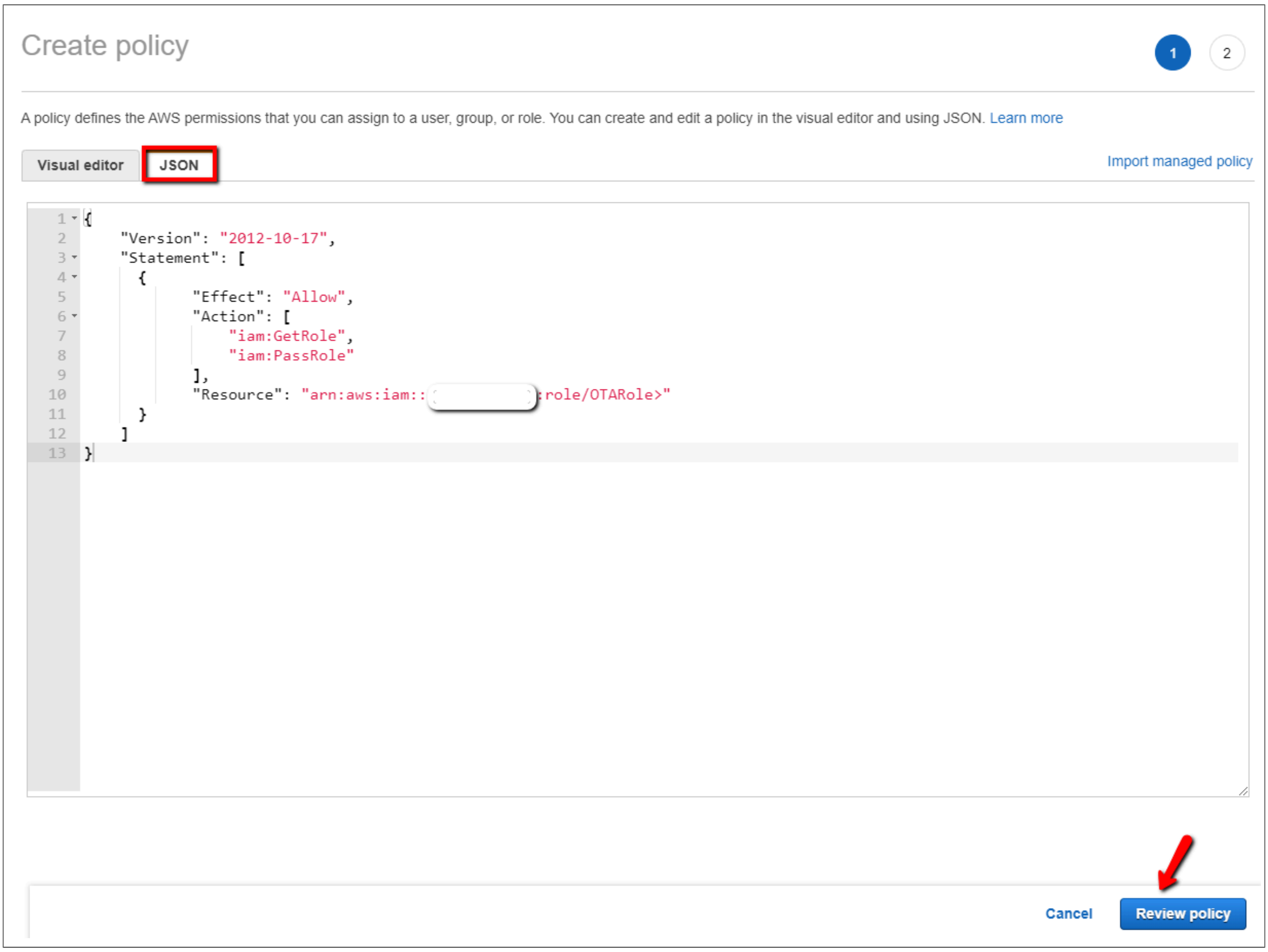
Note: *To obtain account ID, select account name in Web page menu bar and select* ***My account*** *from the drop-down menu. Make note of the* ***Account ID*** *under* ***Account Settings****.*



Account Settings

Account Id:	[redacted]
Seller:	AWS Inc.
Account Name:	[redacted]
Password:	*****

4. Choose **Review policy**.

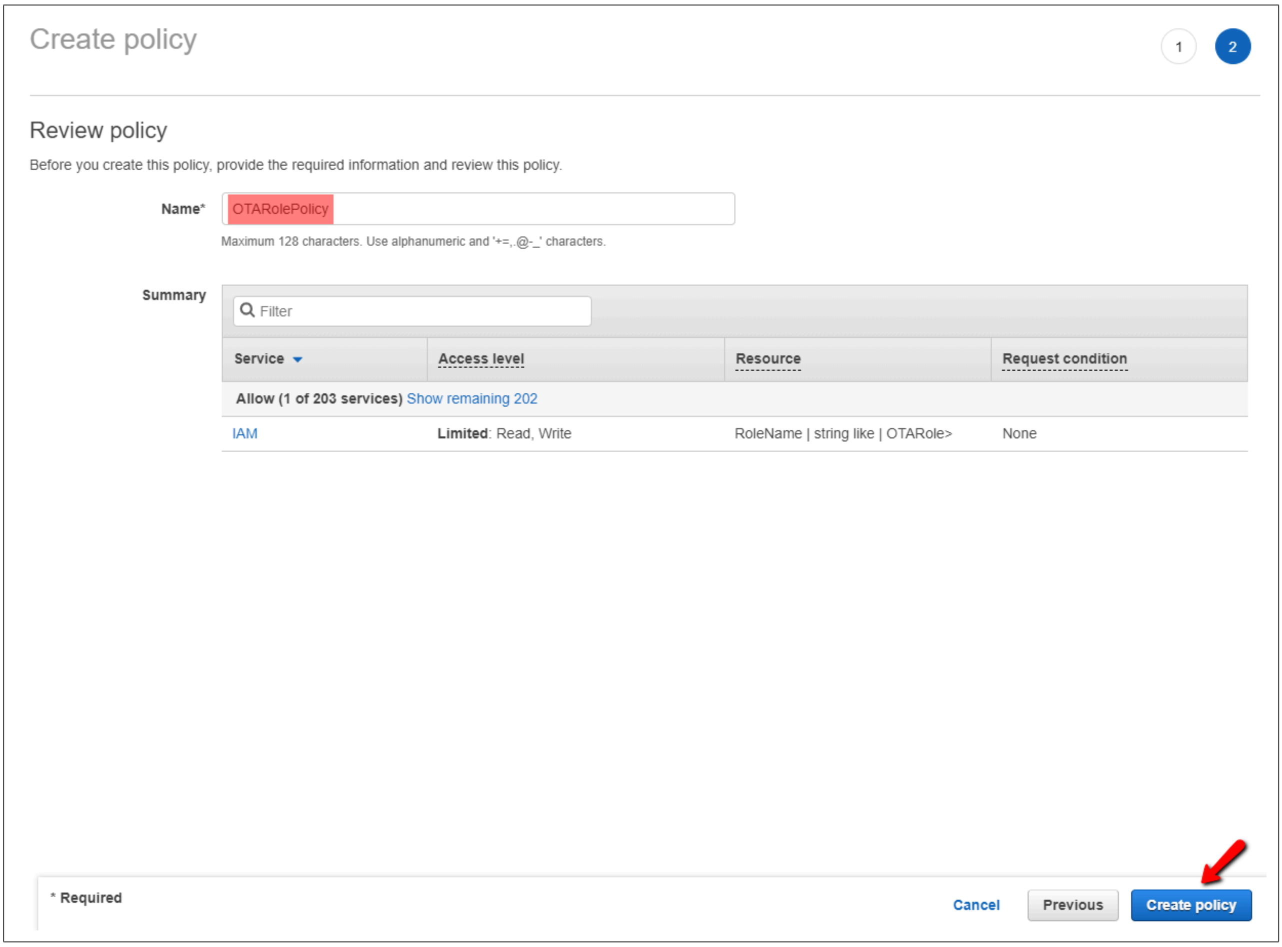


Create policy

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more

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

5. Enter a name for the policy, and then choose **Create policy**.



Create policy

Review policy

Before you create this policy, provide the required information and review this policy.

Name* OTARolePolicy

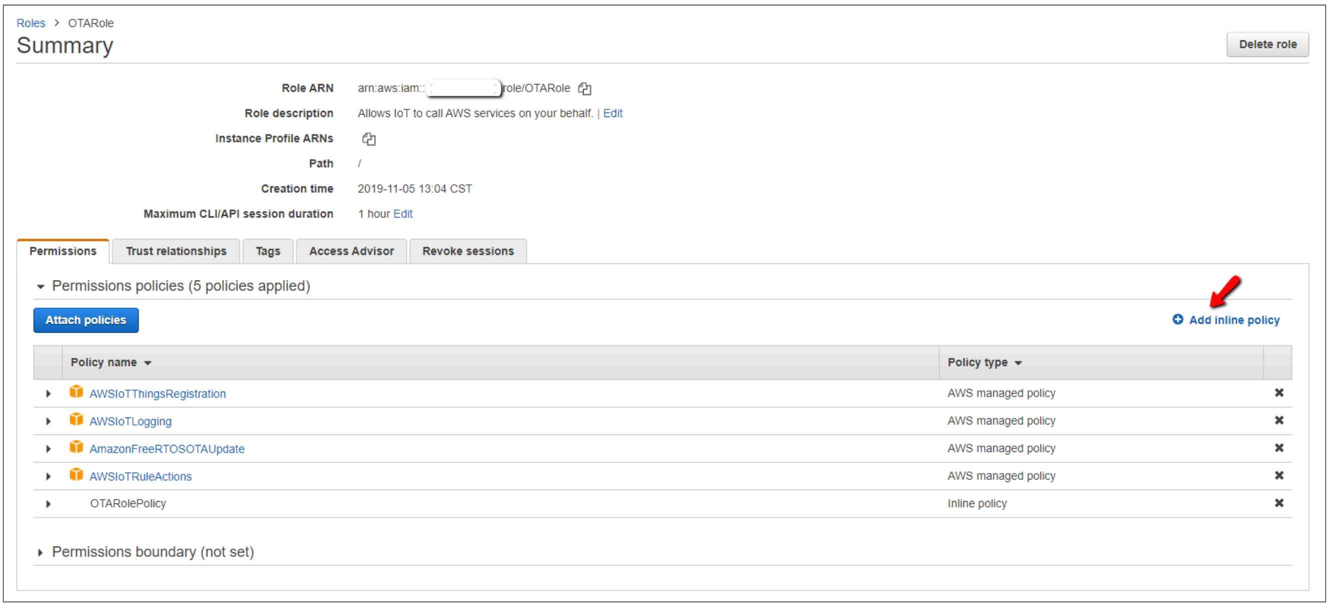
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Summary

Service	Access level	Resource	Request condition
Allow (1 of 203 services) Show remaining 202			
IAM	**Limited**: Read, Write	RoleName | string like | OTARole>	None

2.2.4 To add the required Amazon S3 permissions to your OTA service role

1. In the search box on the IAM console page, enter the name of your role, and then choose it from the list.
2. Choose **Add inline policy**.



Roles > OTARole

Summary

Delete role

Role ARN	arn:aws:iam::[redacted]role/OTARole
Role description	Allows IoT to call AWS services on your behalf. | Edit
Instance Profile ARNs	
Path	/
Creation time	2019-11-05 13:04 CST
Maximum CLI/API session duration	1 hour Edit

Permissions policies (5 policies applied)

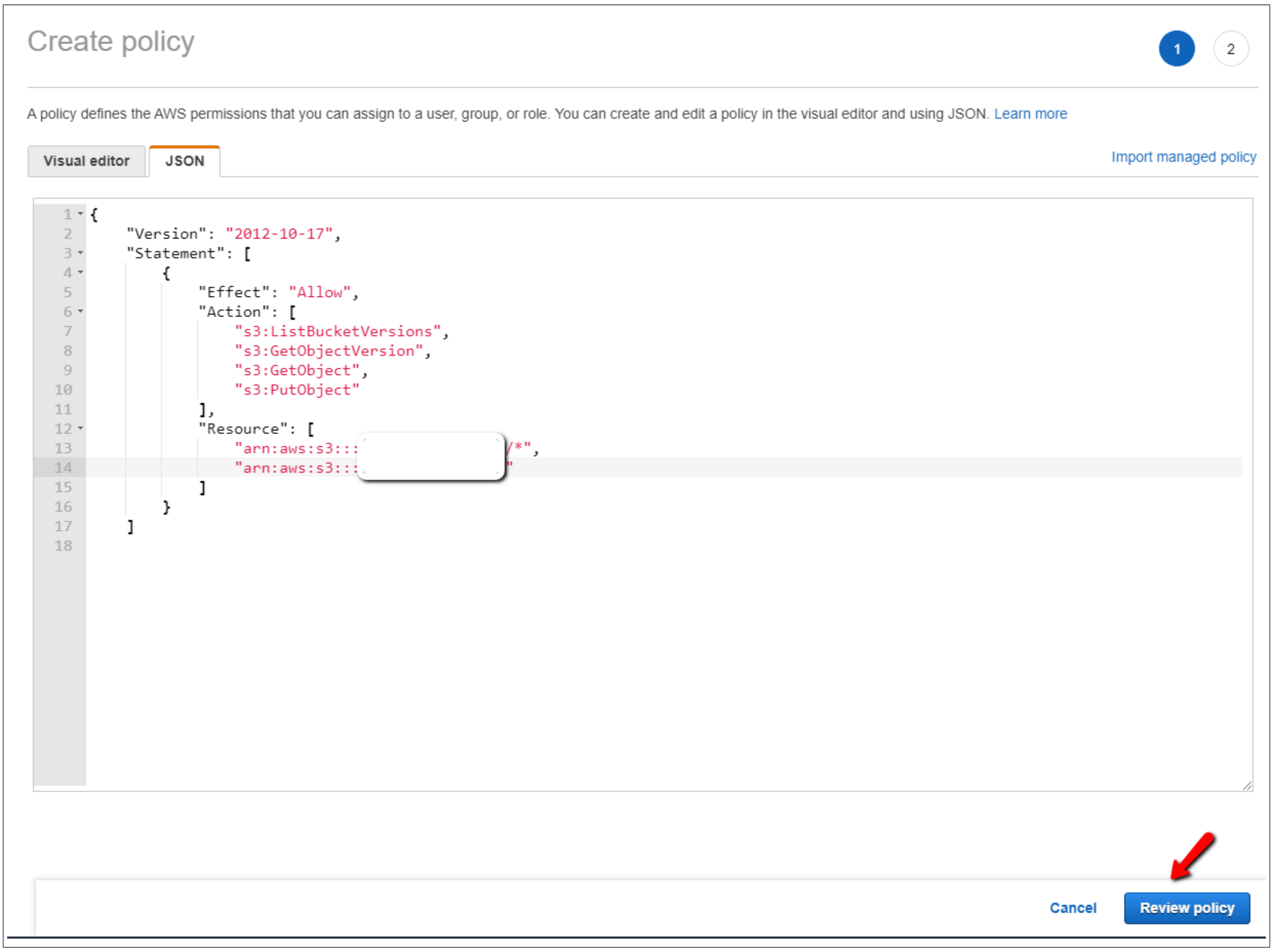
Policy name	Policy type	
AWSIoTThingsRegistration	AWS managed policy	✖
AWSIoTLogging	AWS managed policy	✖
AmazonFreeRTOSOTAUpdate	AWS managed policy	✖
AWSIoTRuleActions	AWS managed policy	✖
OTARolePolicy	Inline policy	✖

Permissions boundary (not set)

3. Choose the **JSON** tab.
   Copy and paste the following policy document into the box:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "s3:ListBucketVersions",
            "s3:GetObjectVersion",
        "s3:GetObject",
        "s3:PutObject"
         ],
         "Resource": [
        "arn:aws:s3:::<example-bucket>/*"
        "arn:aws:s3:::<example-bucket>"
    ]
      }
    ]
}
```

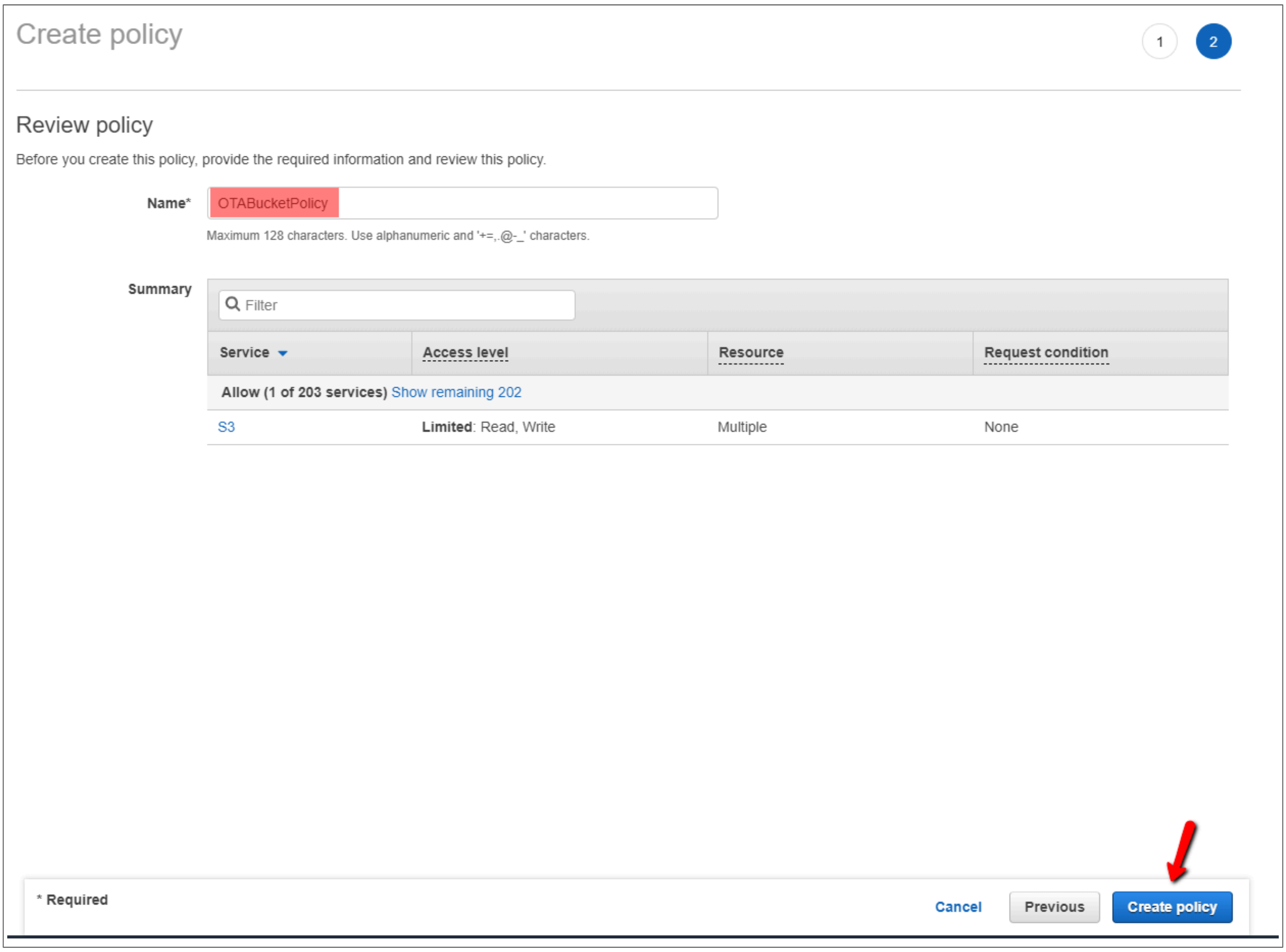
   This policy grants your OTA service role permission to read Amazon S3 objects. Make sure that you replace *<example-bucket>* with the name of your bucket.
4. Choose **Review policy**.



A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucketVersions",
                "s3:GetObjectVersion",
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::[redacted]/*",
                "arn:aws:s3:::"
            ]
        }
    ]
}
```

5. Enter a name for the policy, and then choose **Create policy**.



Create policy

Review policy

Before you create this policy, provide the required information and review this policy.

Name* OTABucketPolicy

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

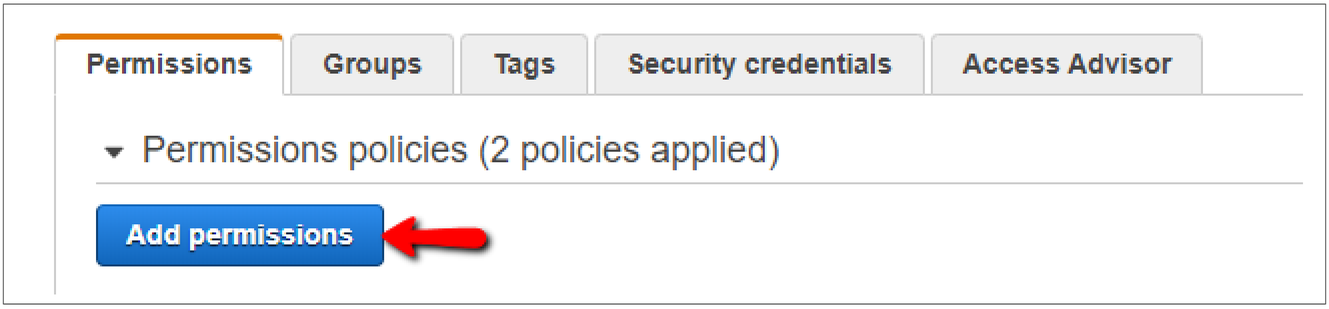
Summary

Service	Access level	Resource	Request condition
Allow (1 of 203 services) Show remaining 202			
S3	**Limited**: Read, Write	Multiple	None

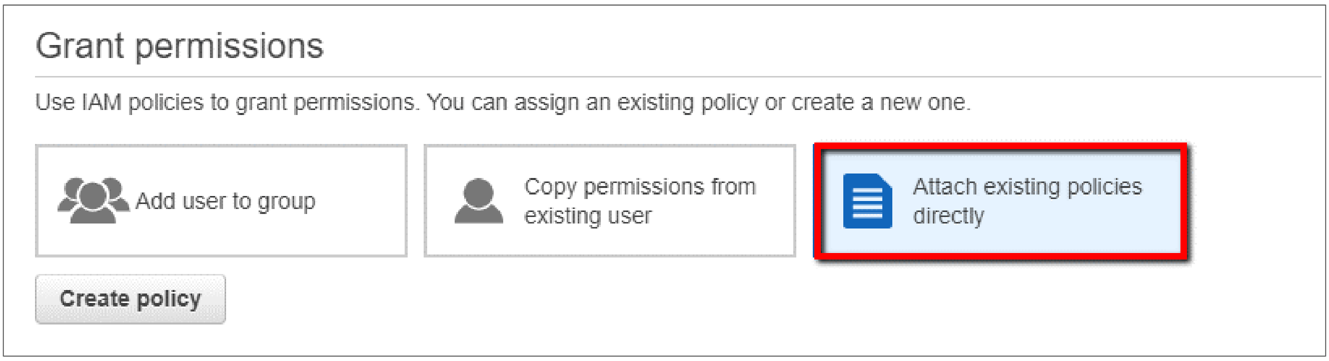
* Required

2.3 Create an OTA user policy

1. Open the https://console.aws.amazon.com/iam/ console.
2. In the navigation pane, choose **Users**.
3. Choose your IAM user from the list.
4. Choose **Add permissions**.



5. Choose **Attach existing policies directly**.

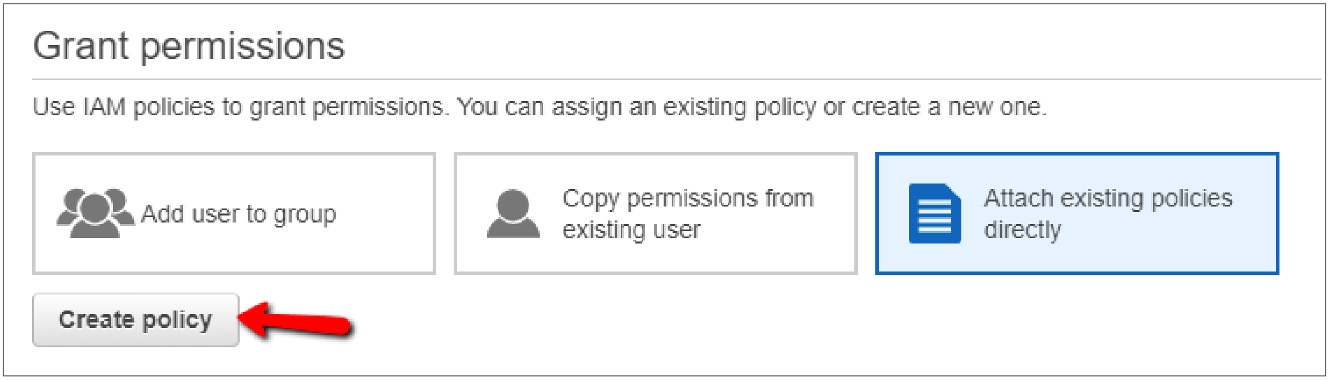


Grant permissions

Use IAM policies to grant permissions. You can assign an existing policy or create a new one.

Add user to group
Copy permissions from existing user
Attach existing policies directly
Create policy

6. Choose **Create policy**.



Grant permissions

Use IAM policies to grant permissions. You can assign an existing policy or create a new one.

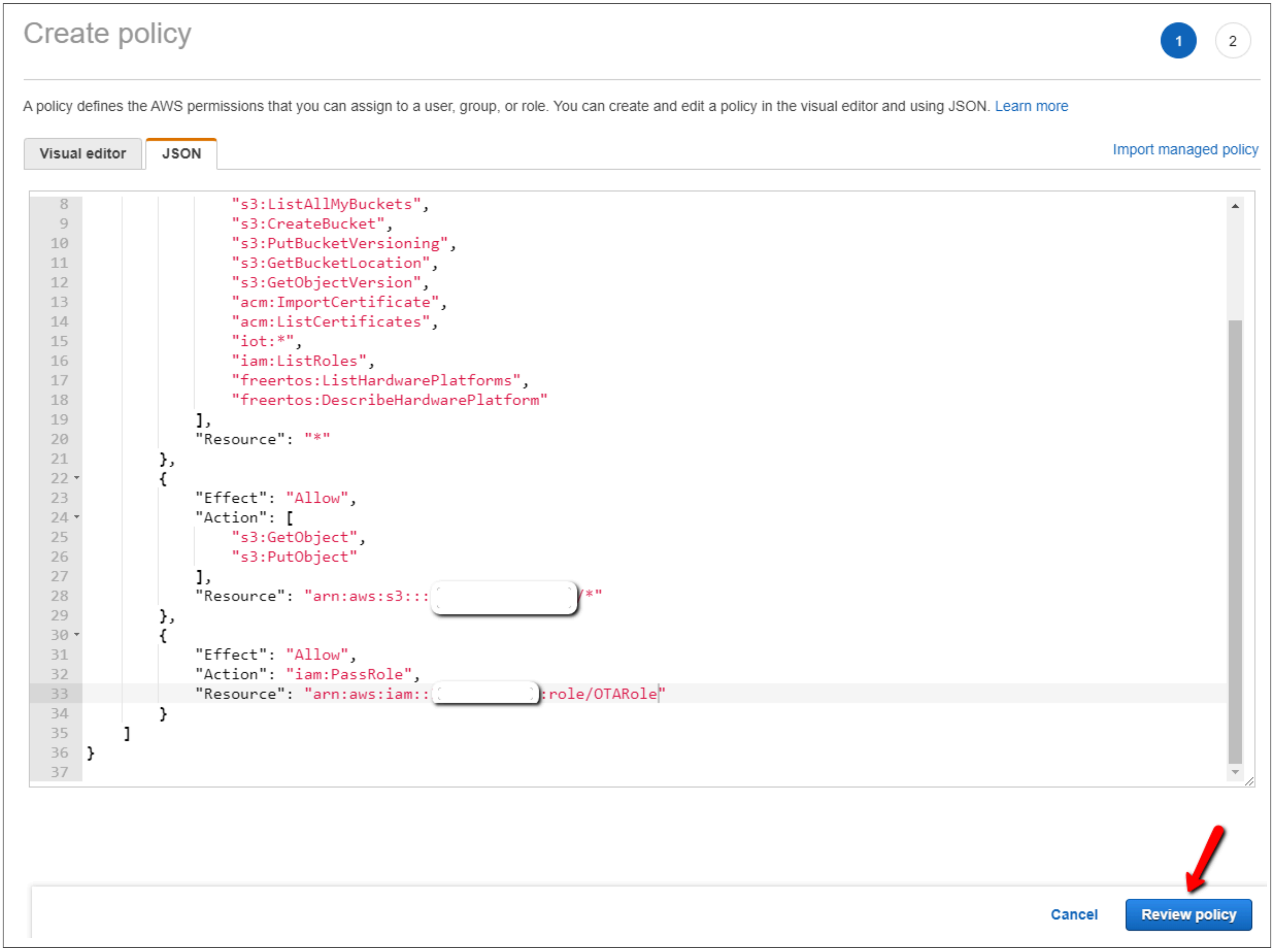
Add user to group
Copy permissions from existing user
Attach existing policies directly
Create policy

Choose the **JSON** tab, and copy and paste the following policy document into the policy editor:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "s3:ListBucket",
      "s3:ListAllMyBuckets",
      "s3:CreateBucket",
      "s3:PutBucketVersioning",
      "s3:GetBucketLocation",
            "s3:GetObjectVersion",
      "acm:ImportCertificate",
      "acm:ListCertificates",
      "iot:*",
      "iam:ListRoles",
      "freertos:ListHardwarePlatforms",
      "freertos:DescribeHardwarePlatform"
         ],
         "Resource": "*"
      }
    {
         "Effect": "Allow",
         "Action": [
      "s3:GetObject",
      "s3:PutObject"
         ],
         "Resource": "arn:aws:s3:::<example-bucket>/*"
      }
    {
         "Effect": "Allow",
         "Action": "iam:PassRole",
         "Resource": "arn:aws:iam::<your-account-id>:role/<role-name>"
      }
    ]
}
```

Replace <*example-bucket*> with the name of the Amazon S3 bucket where your OTA update firmware image is stored. Replace <*your-account-id*> with your AWS account ID. You can find your AWS account ID in the upper right of the console. When you enter your account ID, remove any dashes (-). Replace <*role-name*> with the name of the IAM service role that you created.

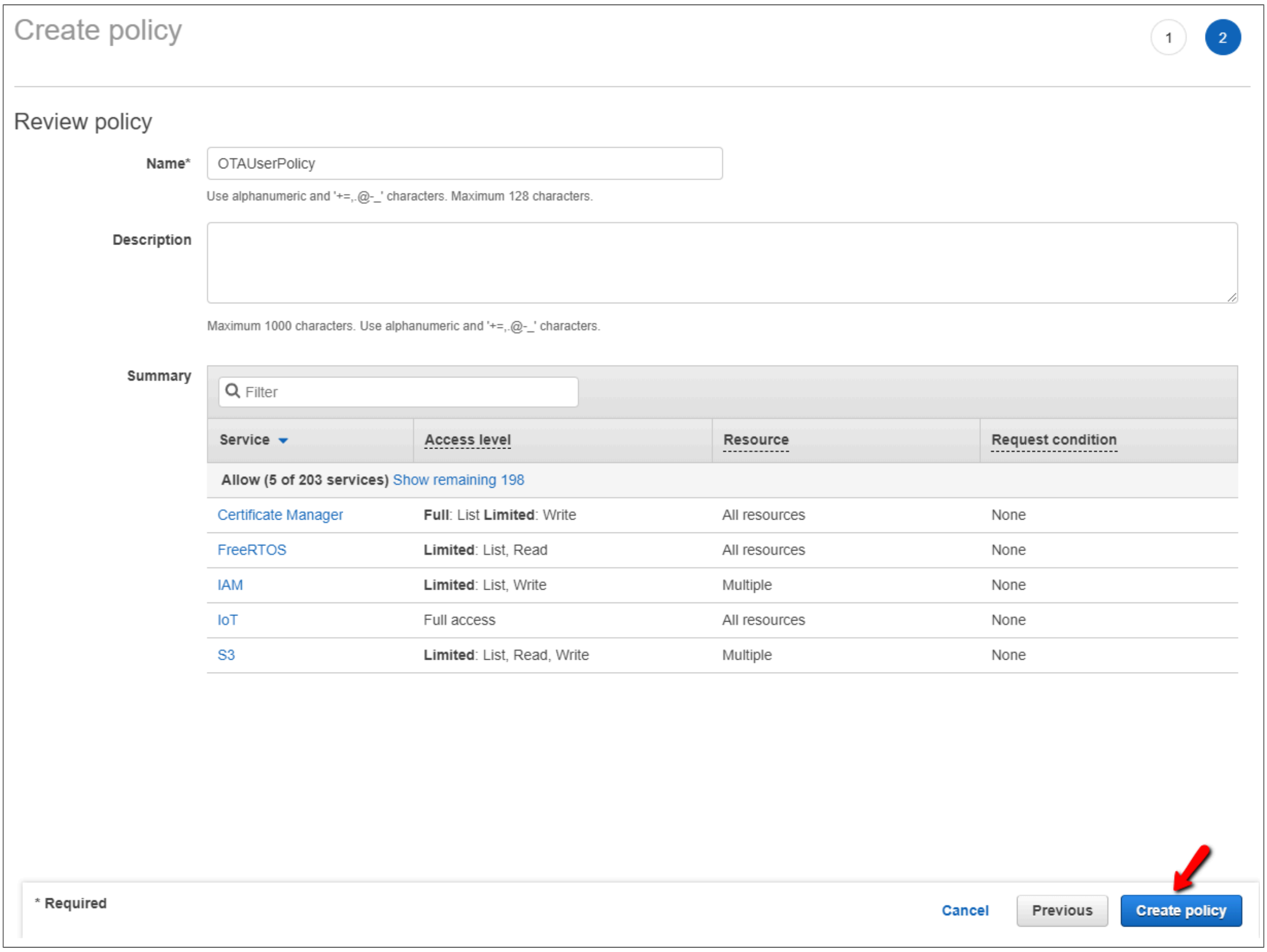
1. Choose **Review policy**.



A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more

```
                "s3:ListAllMyBuckets",
                "s3:CreateBucket",
                "s3:PutBucketVersioning",
                "s3:GetBucketLocation",
                "s3:GetObjectVersion",
                "acm:ImportCertificate",
                "acm:ListCertificates",
                "iot:*",
                "iam:ListRoles",
                "freertos:ListHardwarePlatforms",
                "freertos:DescribeHardwarePlatform"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::[redacted]/*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::[redacted]:role/OTARole"
        }
    ]
}
```

2. Enter a name for your new OTA user policy, and then choose **Create policy**.



Review policy

Name* OTAUserPolicy

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

Description

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Summary

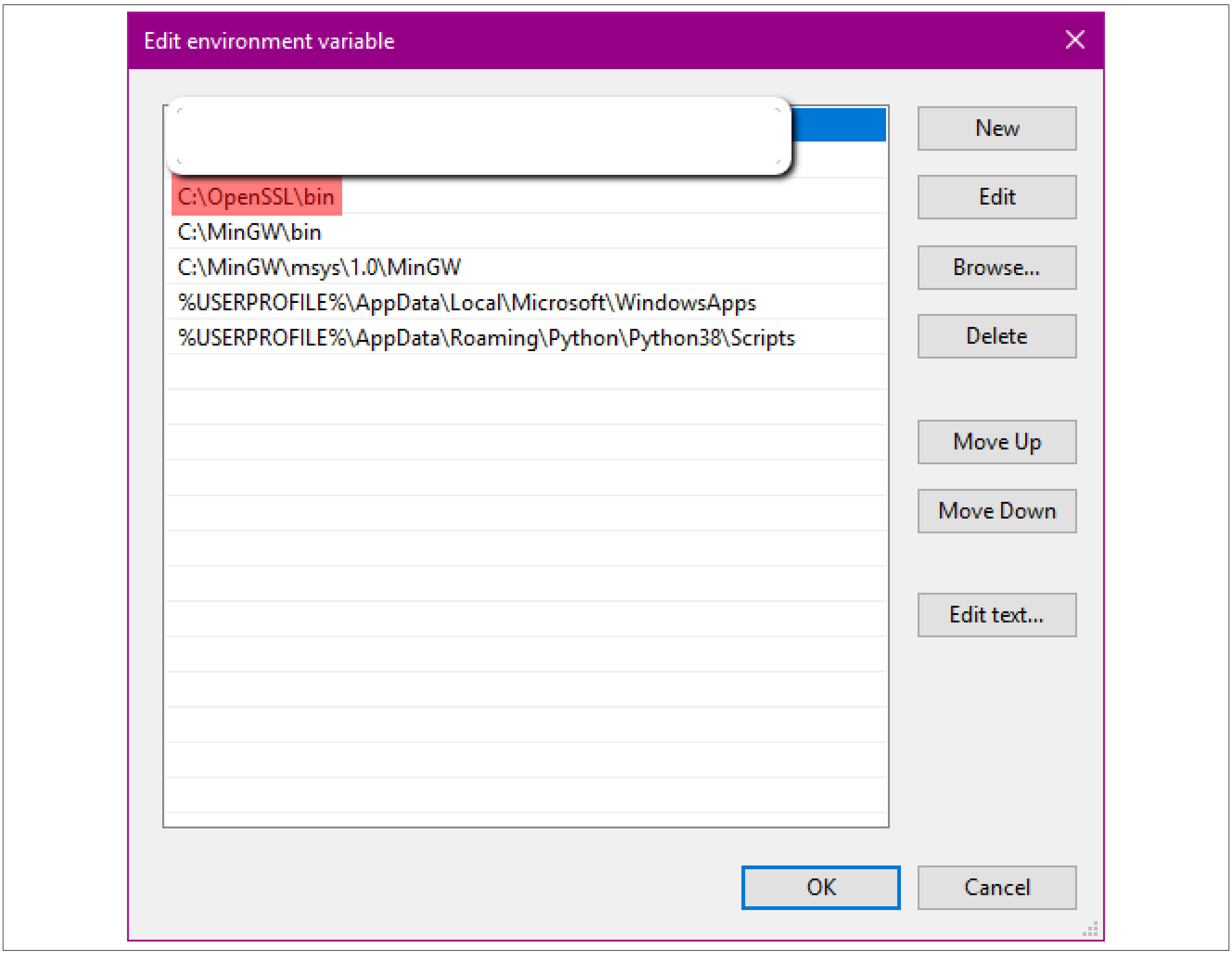
Service	Access level	Resource	Request condition
Allow (5 of 203 services) Show remaining 198			
Certificate Manager	**Full**: List **Limited**: Write	All resources	None
FreeRTOS	**Limited**: List, Read	All resources	None
IAM	**Limited**: List, Write	Multiple	None
IoT	Full access	All resources	None
S3	**Limited**: List, Read, Write	Multiple	None

* Required

2.4 Windows prerequisites

2.4.1 OpenSSL

1. Install OpenSSL
2. Modify the system environment variable path and add your OpenSSL bin directory

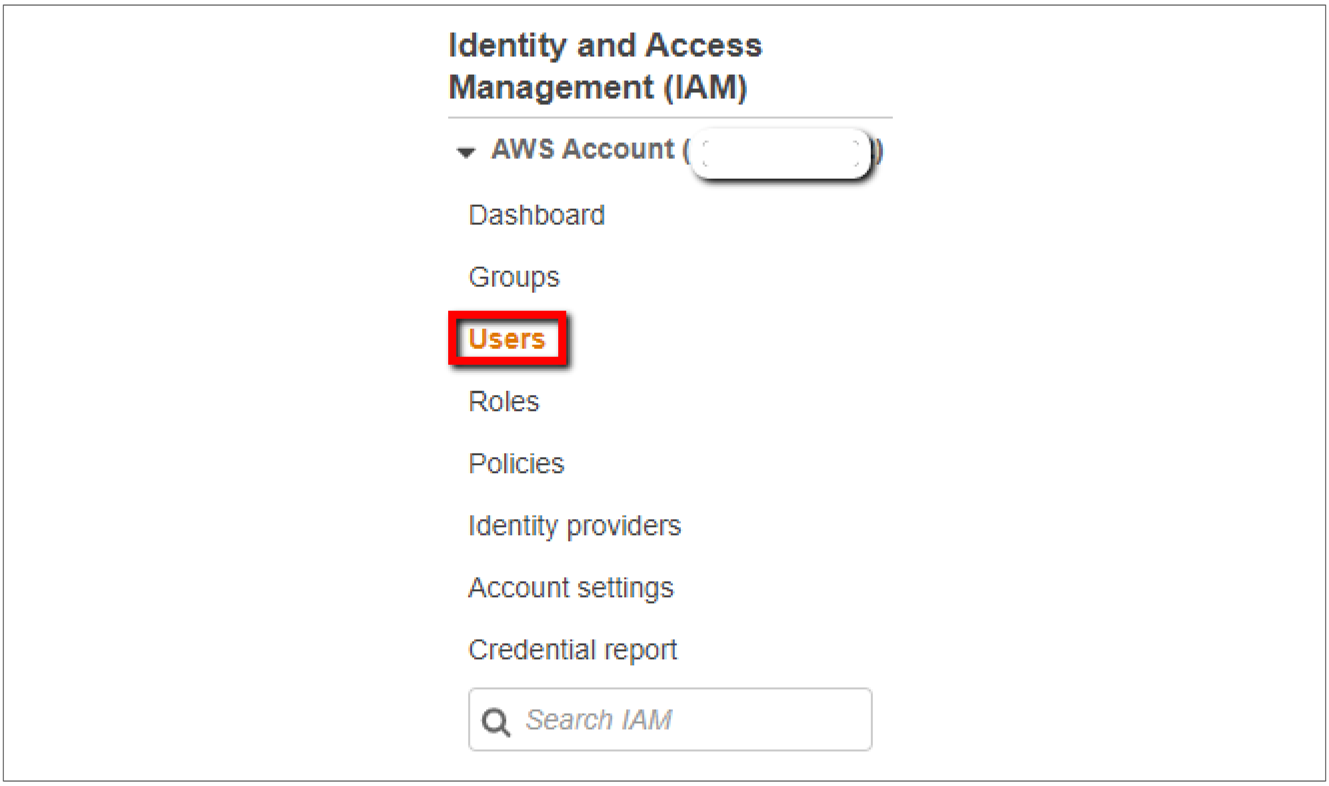


Edit environment variable
C:\OpenSSL\bin
C:\MinGW\bin
C:\MinGW\msys\1.0\MinGW
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
%USERPROFILE%\AppData\Roaming\Python\Python38\Scripts
New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...
OK
Cancel

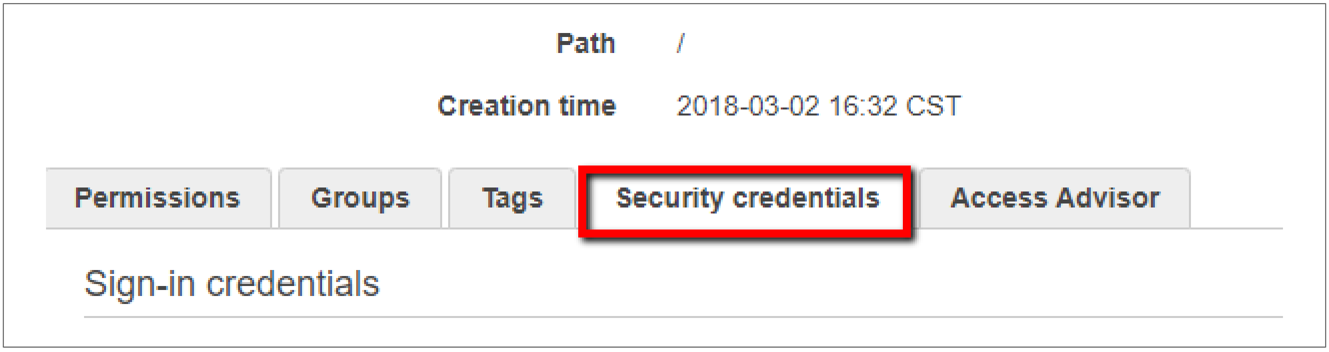
Make sure that openssl gets assigned to the OpenSSL executable in your command prompt or terminal environment.

2.4.2 Install the AWS CLI

1. Follow the instructions for AWS CLI bundler installer https://docs.aws.amazon.com/cli/latest/userguide/install-windows.html#install-msi-on-windows
2. Go to the IAM console https://console.aws.amazon.com/iam/
3. In the navigation pane, choose **Users**.



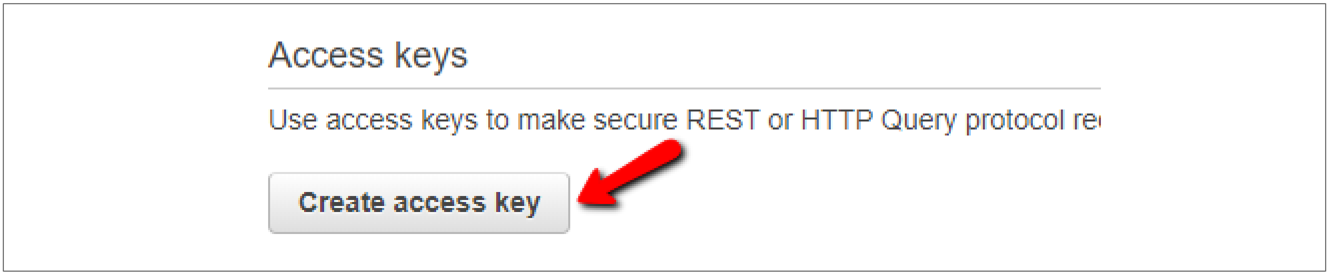
4. Choose your IAM user account.
5. Select **Security credentials**



Path /

Creation time 2018-03-02 16:32 CST

6. In the **Access keys** section, choose **Create access key**.

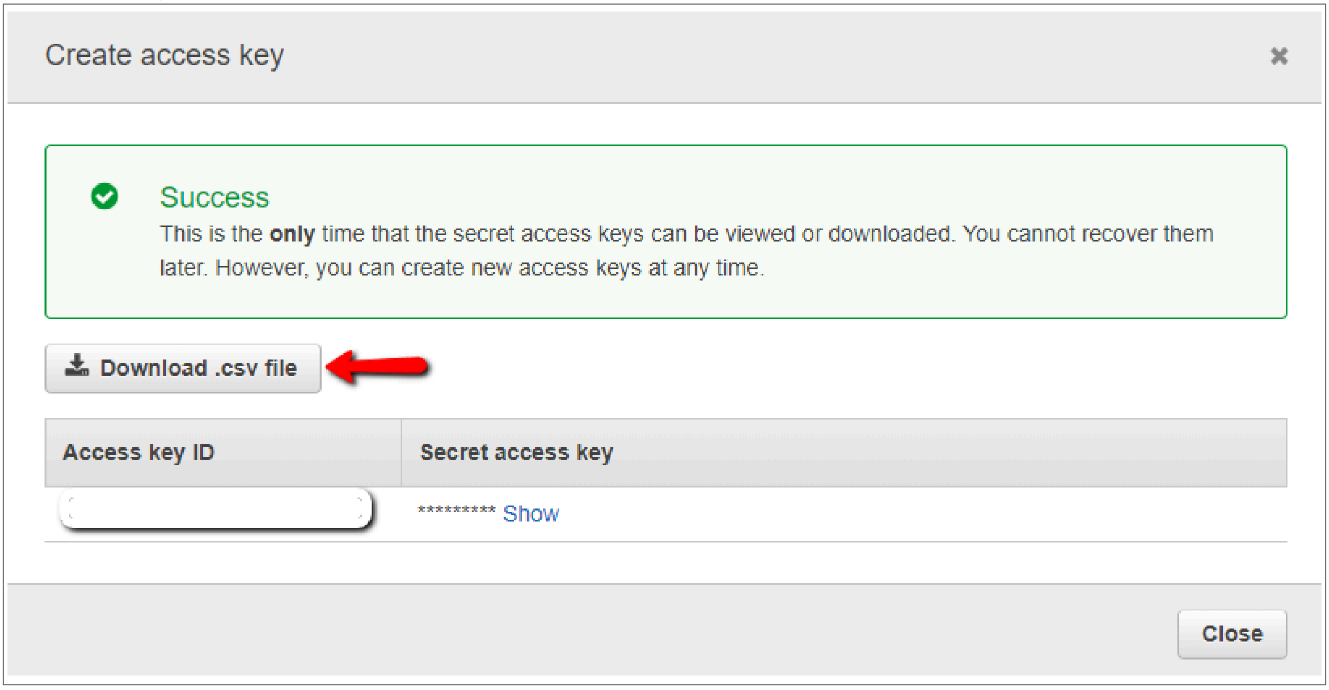


Access keys

Use access keys to make secure REST or HTTP Query protocol re

7. To view the new access key pair, choose **Show**. You will not have access to the secret access key again after this dialog box closes. Your credentials look something like this: *Access key ID: AKIAIOSFODNN7EXAMPLE Secret access key: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY*
8. To download the key pair, choose **Download .csv** file. Store the keys in a secure location. You will not have access to the secret access key again after this dialog box closes. Keep the keys confidential to protect your AWS account and never email them. Do not share them outside your organization, even if an inquiry

appears to come from AWS or Amazon.com. No one who legitimately represents Amazon asks you for your secret key.



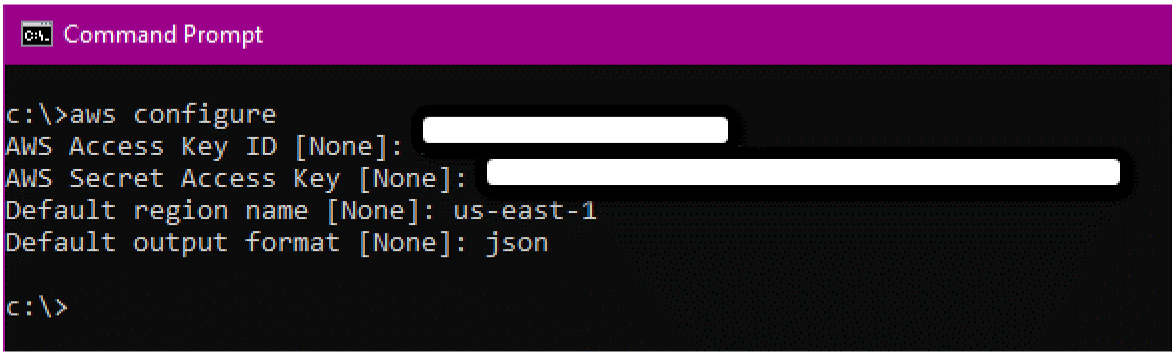
Create access key

Success

This is the **only** time that the secret access keys can be viewed or downloaded. You cannot recover them later. However, you can create new access keys at any time.

Access key ID	Secret access key
[redacted]	********* Show

9. After you download the .csv file, choose **Close**. When you create an access key, the key pair is active by default, and you can use the pair right away.
10. For general use, the aws configure command is the fastest way to set up your AWS CLI installation



```
c:\>aws configure
AWS Access Key ID [None]: [redacted]
AWS Secret Access Key [None]: [redacted]
Default region name [None]: us-east-1
Default output format [None]: json

c:\>
```

2.5 Creating a code-signing certificate

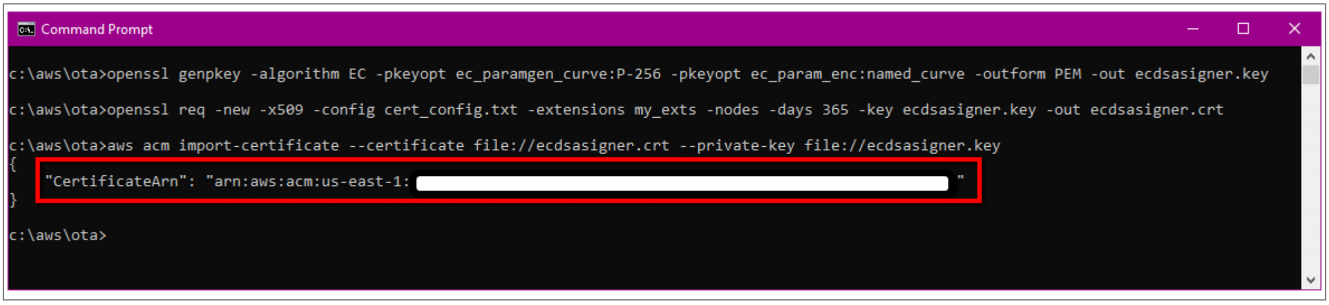
1. In your working directory, use the following text to create a file named cert_config.txt. Replace `test_signer@amazon.com` with your email address.

```
[ req ]
prompt                 = no
distinguished_name = my_dn

[ my_dn ]
commonName = test_signer@amazon.com

[ my_exts ]
keyUsage           = digitalSignature
extendedKeyUsage = codeSigning
```

2. Using openSSL command line create an ECDSA code-signing private key

```
openssl genpkey-algorithm EC -pkeyopt ec_paramgen_curve:P-256 -pkeyopt
 ec_param_enc:named_curve -outform PEM -out ecdsasigner.key
```

3. Create an ECDSA code-signing certificate: openssl req -new -x509 -config cert_config.txt -extensions my_exts -nodes -days 365 -key ecdsasigner.key -out ecdsasigner.crt
4. Import the code-signing certificate, private key, and certificate chain into AWS Certificate Manager: aws acm import-certificate *--certificate file://ecdsasigner.crt --private-key file://ecdsasigner.key* ***Note:*** *this command displays an ARN for your certificate. Save it locally to use it while creating the OTA update job*.



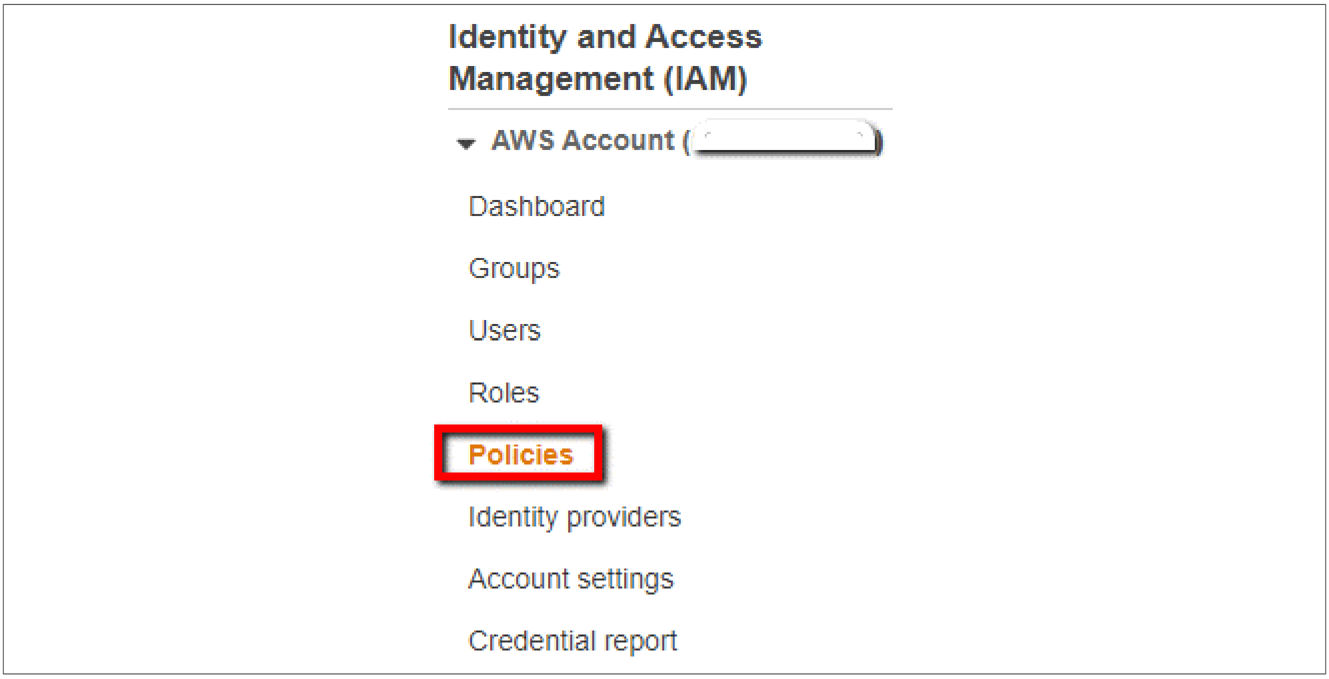
```
c:\aws\ota>openssl genpkey -algorithm EC -pkeyopt ec_paramgen_curve:P-256 -pkeyopt ec_param_enc:named_curve -outform PEM -out ecdsasigner.key

c:\aws\ota>openssl req -new -x509 -config cert_config.txt -extensions my_exts -nodes -days 365 -key ecdsasigner.key -out ecdsasigner.crt

c:\aws\ota>aws acm import-certificate --certificate file://ecdsasigner.crt --private-key file://ecdsasigner.key
{
  "CertificateArn": "arn:aws:acm:us-east-1:[redacted]"
}

c:\aws\ota>
```

5. Get the ECDSA public key from the code signing credentials

```
openssl ec -in ecdsasigner.key -pubout -outform PEM -out ecdsasigner-pub-
key.pem
```

3 Grant access to code signing for AWS IoT

1. Sign in to the https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Policies**.

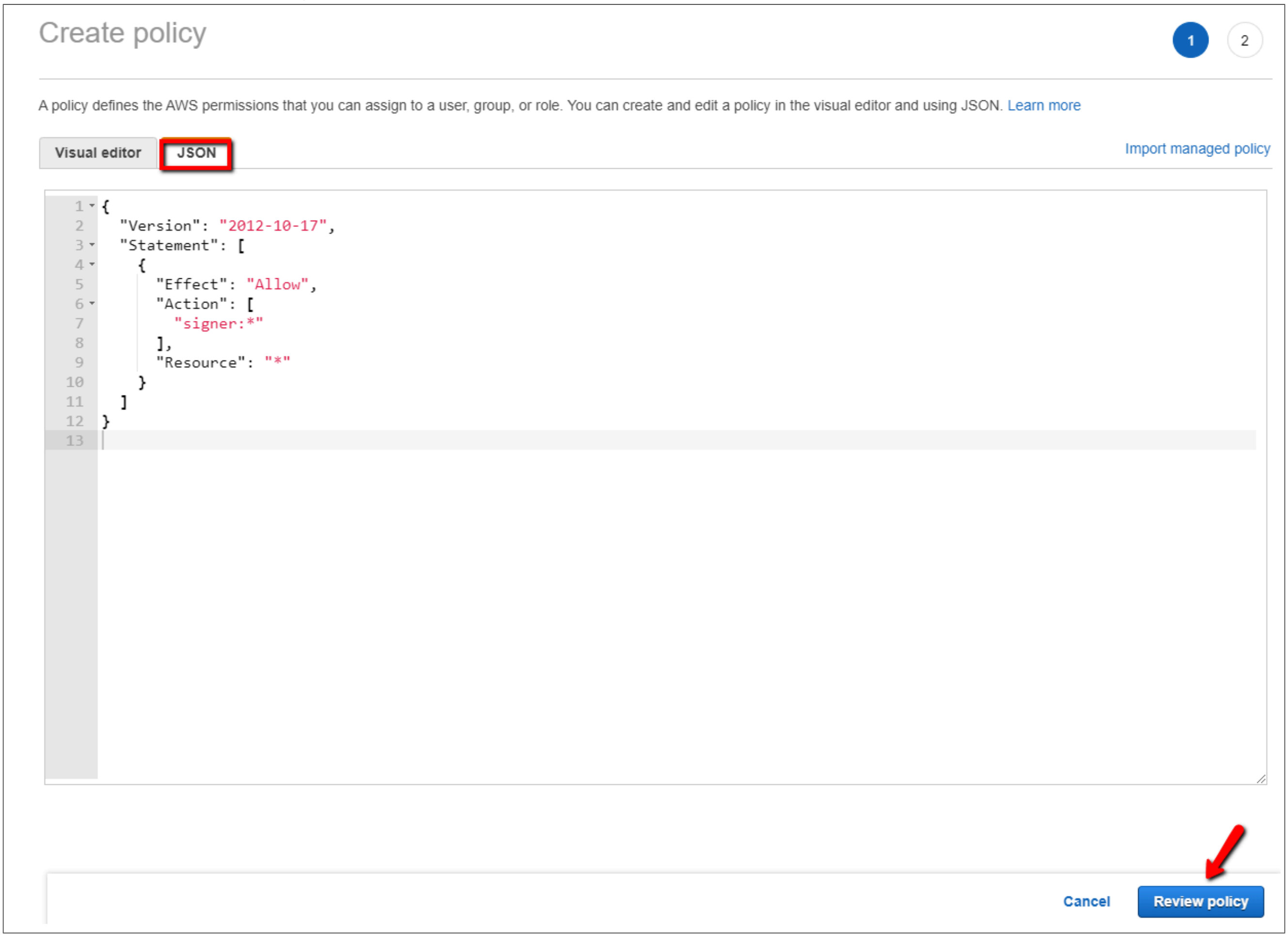


3. Choose **Create Policy**.

4. On the **JSON** tab, copy and paste the following JSON document into the policy editor. This policy allows the IAM user access to all code-signing operations.

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

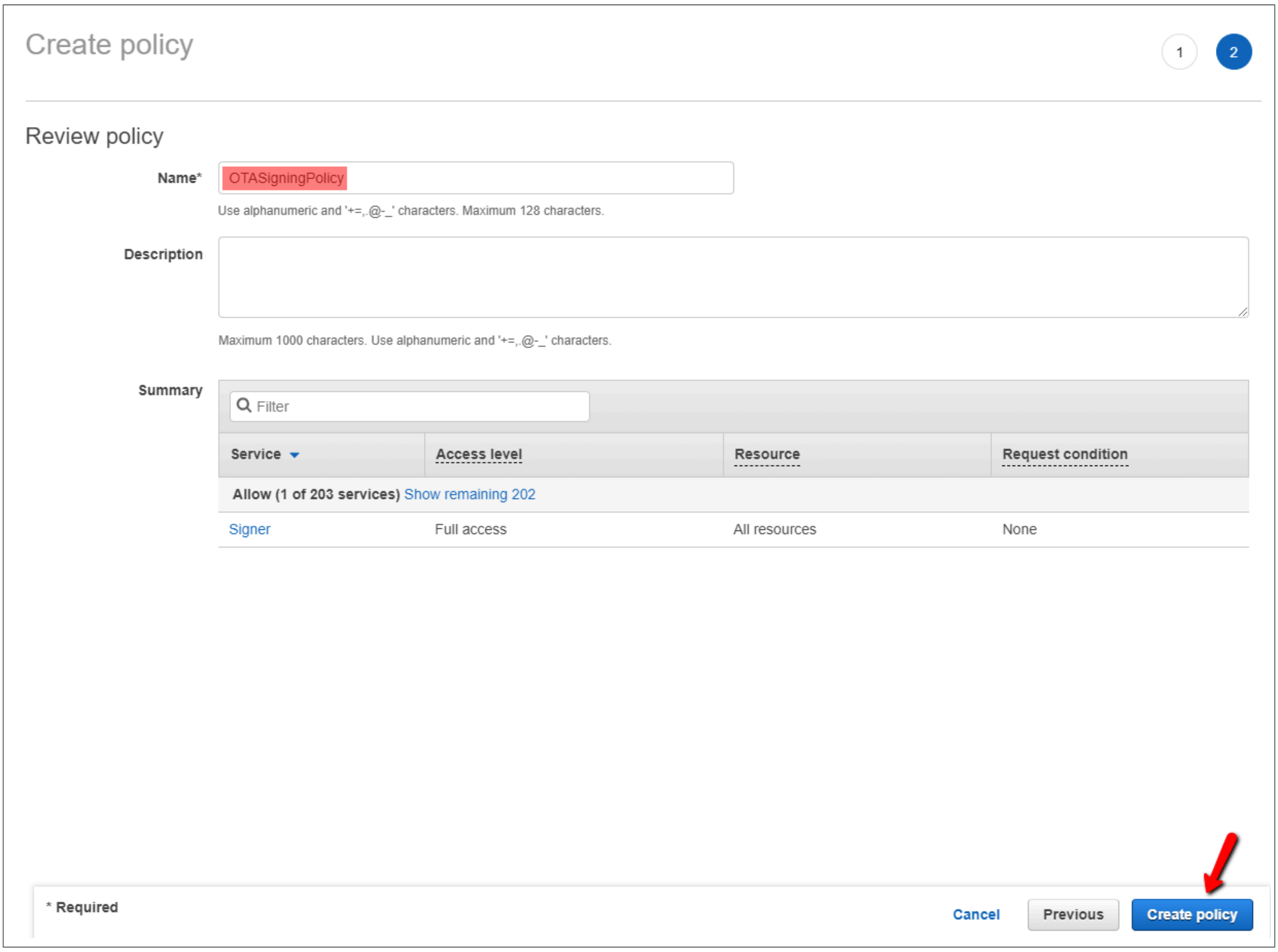
5. Choose **Review policy**.



A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more

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

6. Enter a policy name and description, and then choose **Create policy**.



Create policy

Review policy

Name* OTASigningPolicy

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

Description

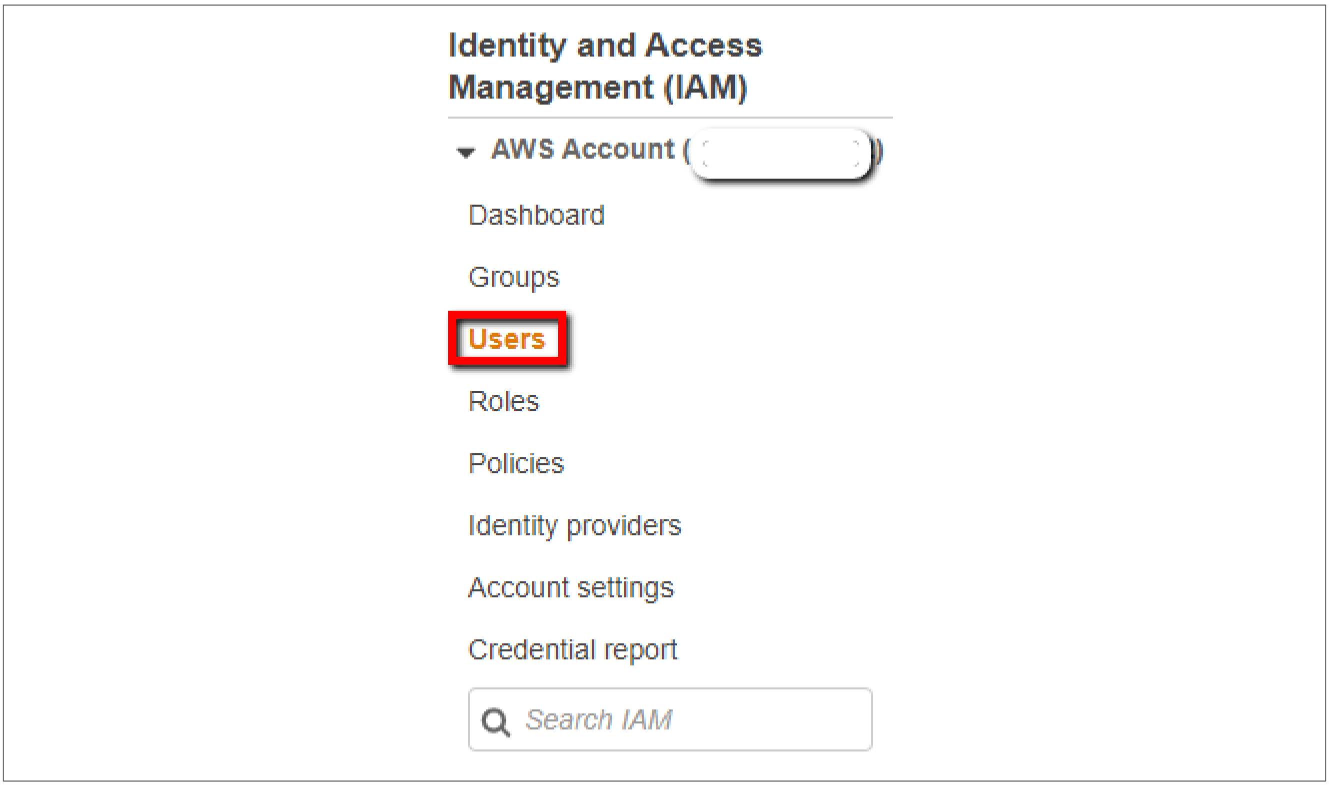
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Summary

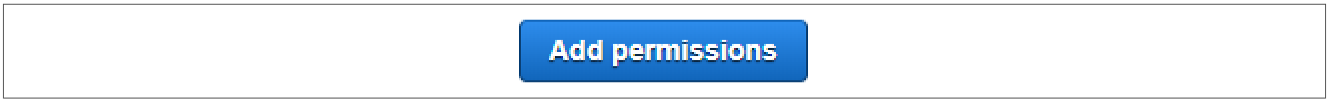
Service	Access level	Resource	Request condition
Allow (1 of 203 services) Show remaining 202			
Signer	Full access	All resources	None

* Required

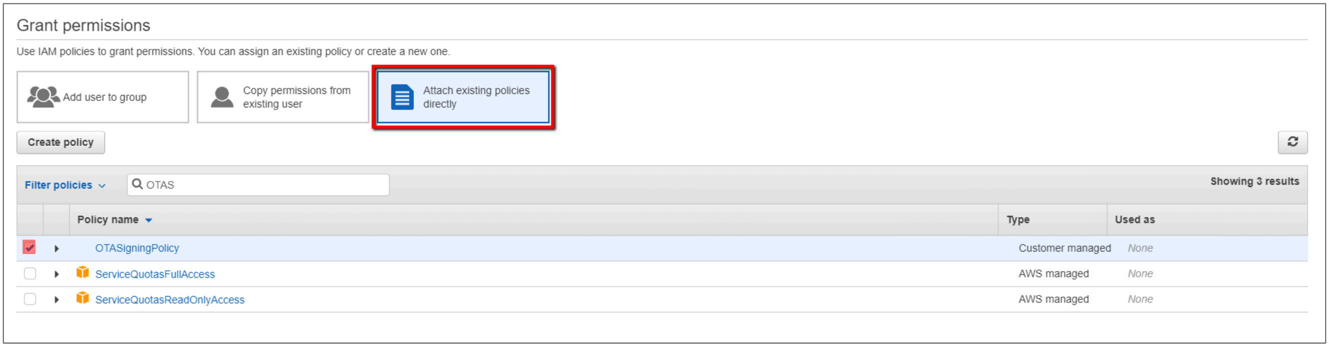
7. In the navigation pane, choose **Users**.



8. Choose your IAM user account.
9. On the **Permissions** tab, choose **Add permissions**.



10. Choose **Attach existing policies directly**, and then select the checkbox next to the code-signing policy you created.



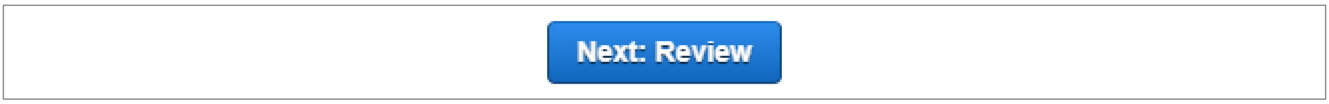
Grant permissions

Use IAM policies to grant permissions. You can assign an existing policy or create a new one.

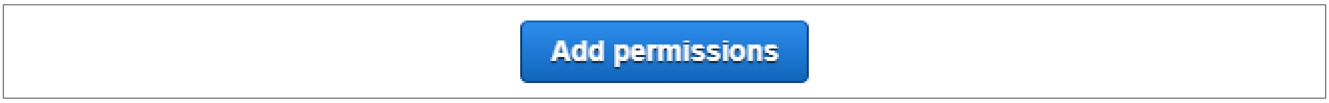
Add user to group
Copy permissions from existing user
Attach existing policies directly
Create policy
Filter policies
OTAS
Showing 3 results

		Policy name	Type	Used as
☑	▸	OTASigningPolicy	Customer managed	None
□	▸	ServiceQuotasFullAccess	AWS managed	None
□	▸	ServiceQuotasReadOnlyAccess	AWS managed	None

11. Choose **Next: Review**.



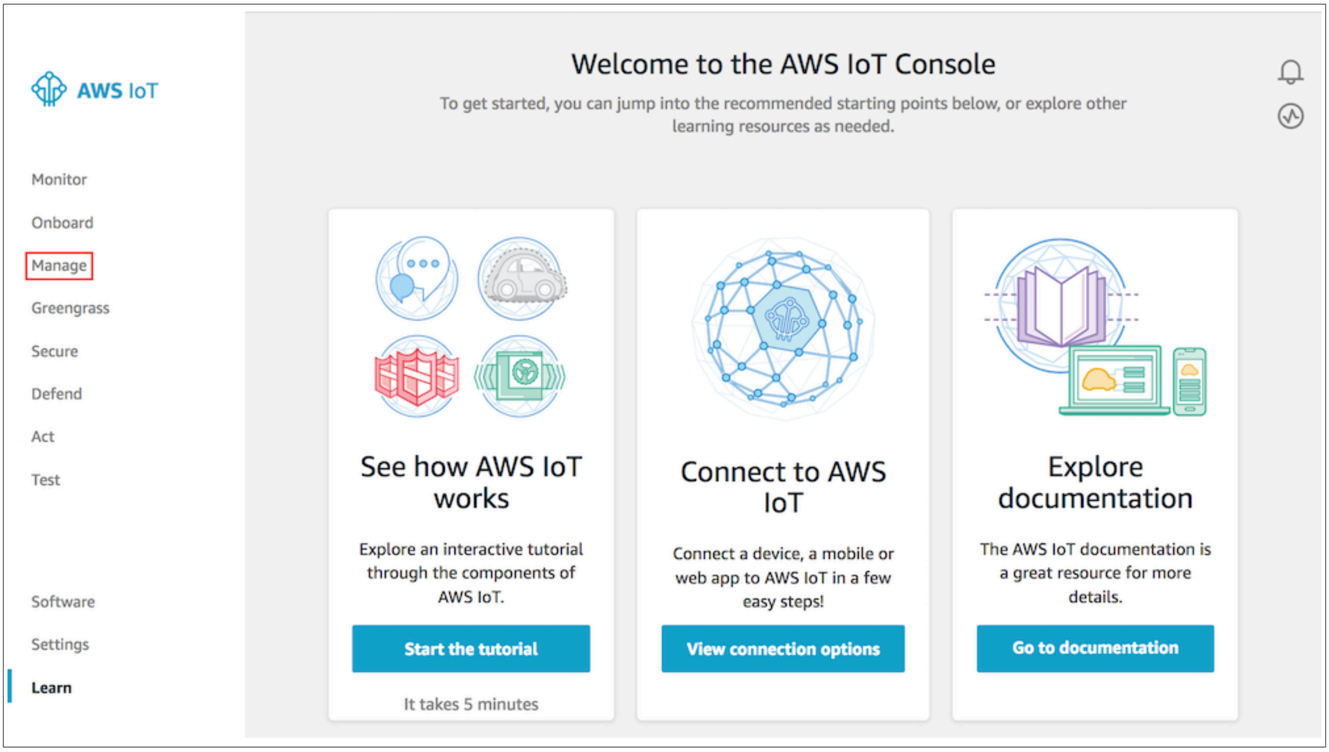
12. Choose **Add permissions**.



4 AWS IoT

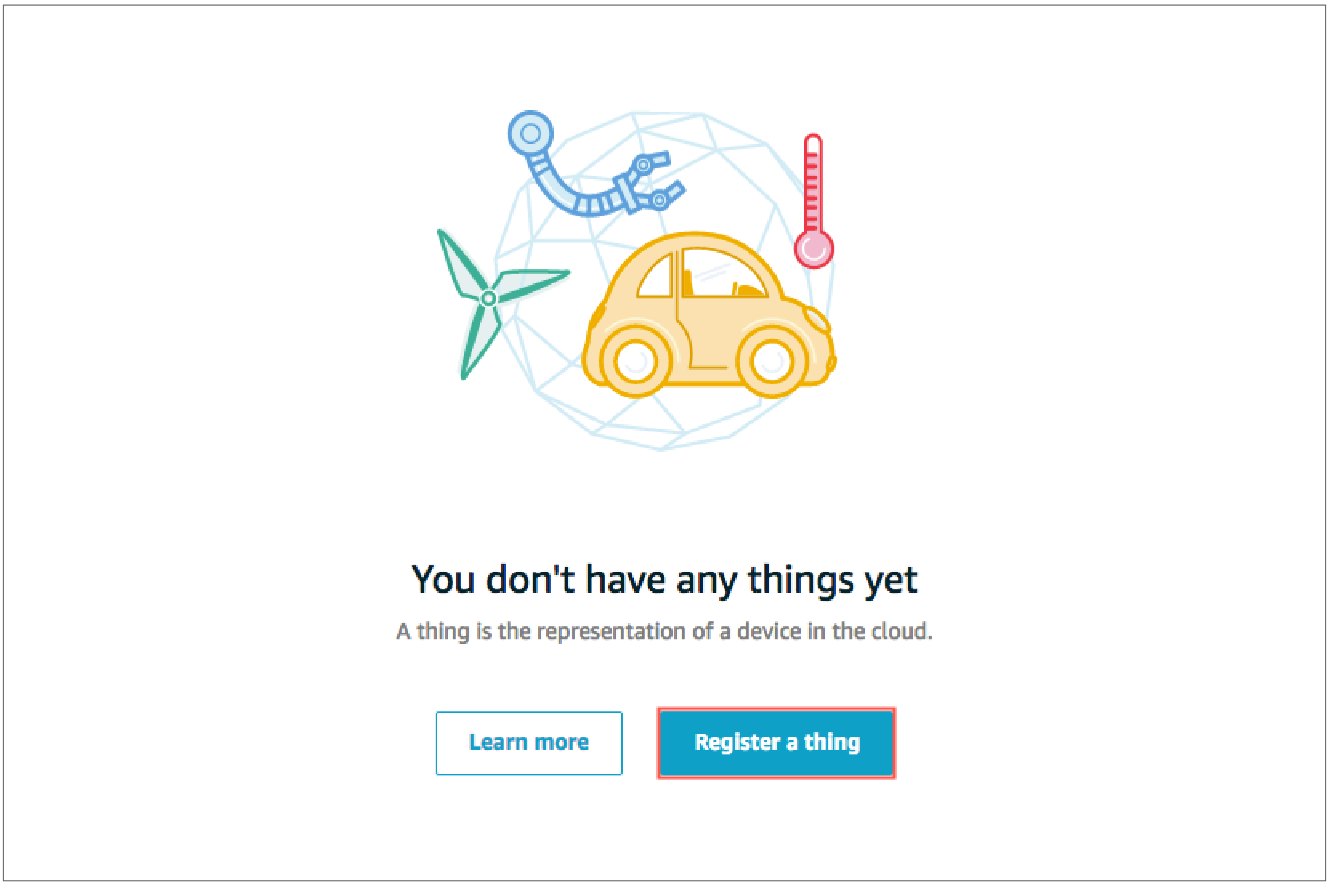
4.1 Create an AWS IoT thing

1. Open the AWS IoT console website https://console.aws.amazon.com/iot/.
2. On the **Welcome to the AWS IoT Console** page, in the navigation pane, choose **Manage**.



AWS IoT
Monitor
Onboard
Manage
Greengrass
Secure
Defend
Act
Test
Software
Settings
Learn
Welcome to the AWS IoT Console
To get started, you can jump into the recommended starting points below, or explore other learning resources as needed.
See how AWS IoT works
Explore an interactive tutorial through the components of AWS IoT.
Start the tutorial
It takes 5 minutes
Connect to AWS IoT
Connect a device, a mobile or web app to AWS IoT in a few easy steps!
View connection options
Explore documentation
The AWS IoT documentation is a great resource for more details.
Go to documentation

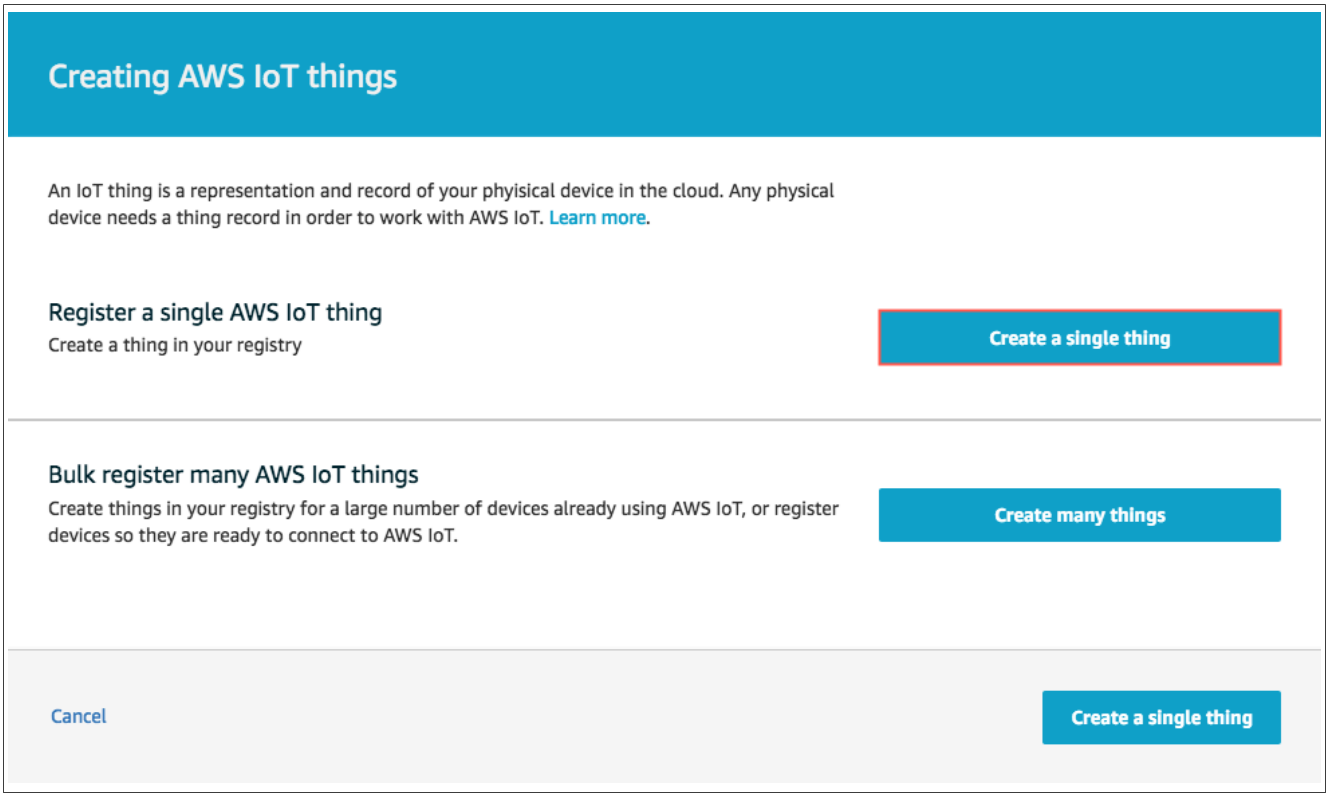
3. On the **You don't have any things yet** page, choose **Register a thing**.



You don't have any things yet

A thing is the representation of a device in the cloud.

4. On the **Creating AWS IoT things** page, choose **Create a single thing**.



Creating AWS IoT things

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

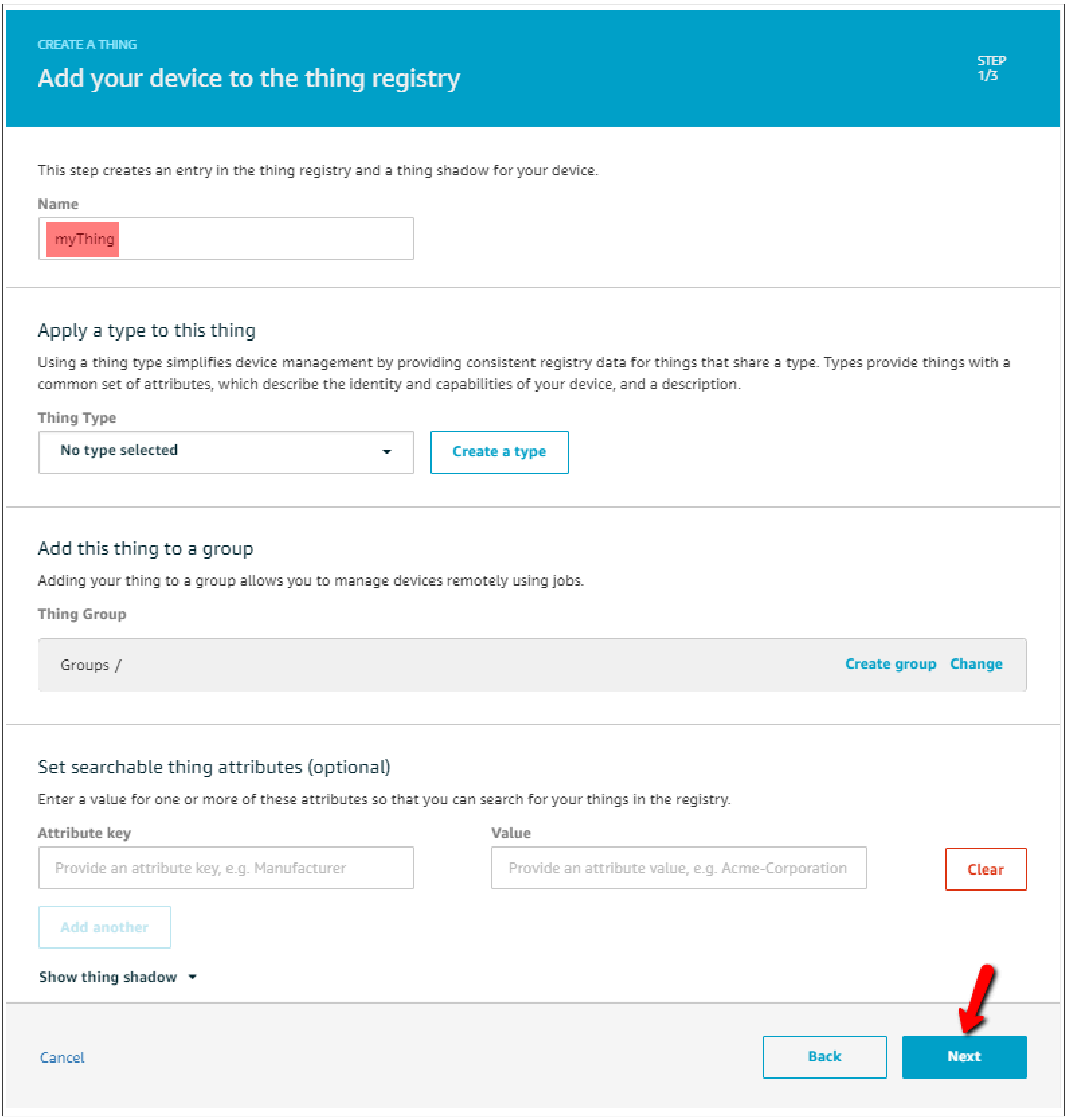
Register a single AWS IoT thing

Create a thing in your registry

Bulk register many AWS IoT things

Create things in your registry for a large number of devices already using AWS IoT, or register devices so they are ready to connect to AWS IoT.

5. On the **Create a thing** page, in the **Name** field, enter a name for your thing, such as MyThing. Choose **Next**.



CREATE A THING

Add your device to the thing registry

STEP 1/3

This step creates an entry in the thing registry and a thing shadow for your device.

Name

myThing

Apply a type to this thing

Using a thing type simplifies device management by providing consistent registry data for things that share a type. Types provide things with a common set of attributes, which describe the identity and capabilities of your device, and a description.

Thing Type

No type selected

Create a type

Add this thing to a group

Adding your thing to a group allows you to manage devices remotely using jobs.

Thing Group

Groups /

Create group Change

Set searchable thing attributes (optional)

Enter a value for one or more of these attributes so that you can search for your things in the registry.

Attribute key

Provide an attribute key, e.g. Manufacturer

Value

Provide an attribute value, e.g. Acme-Corporation

Clear

Add another

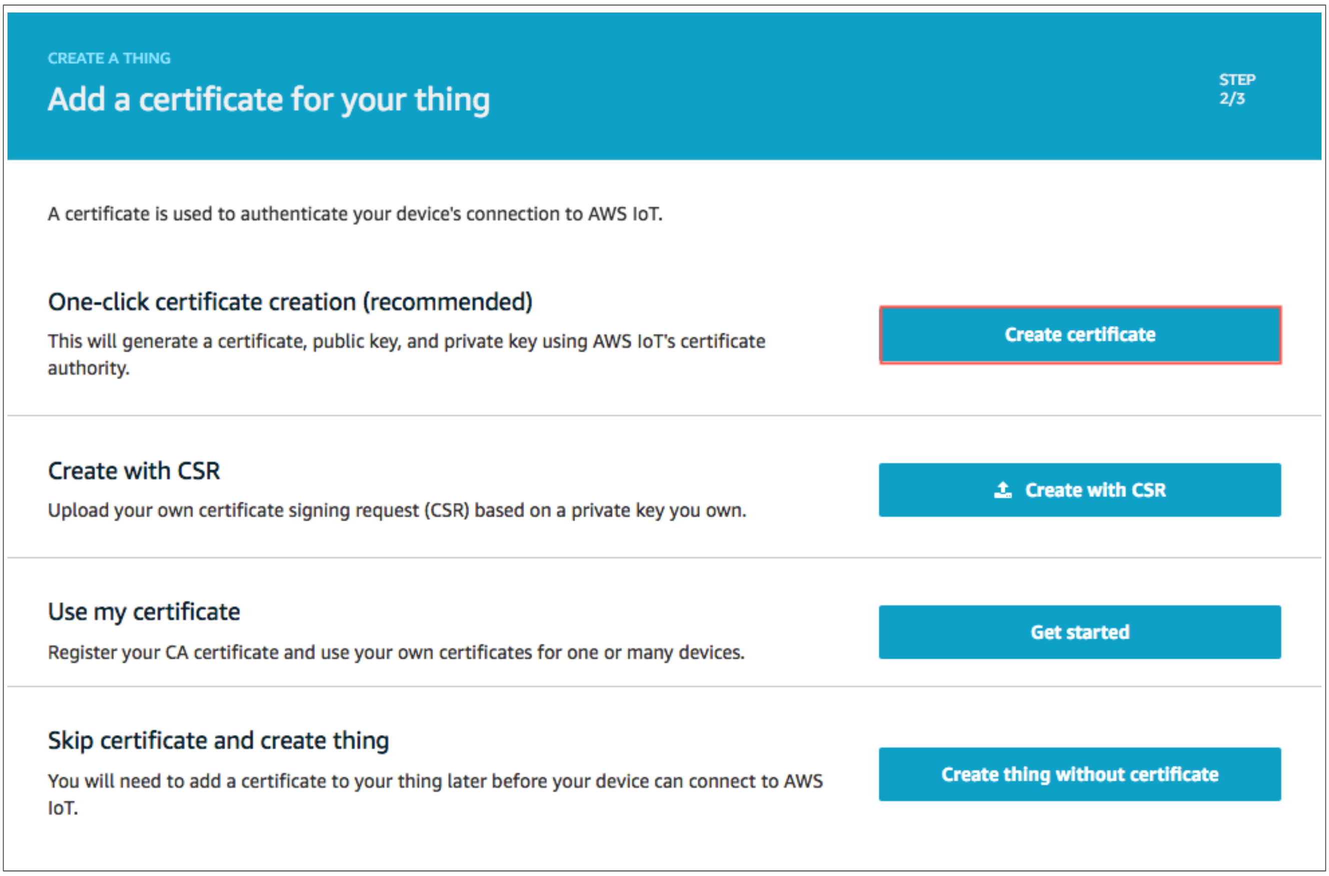
Show thing shadow

Cancel

Back

Next

6. On the **Add a certificate for your thing** page, choose **Create certificate**. This generates an X.509 certificate and a key pair.



CREATE A THING

Add a certificate for your thing

STEP 2/3

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

One-click certificate creation (recommended)

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

Create with CSR

Upload your own certificate signing request (CSR) based on a private key you own.

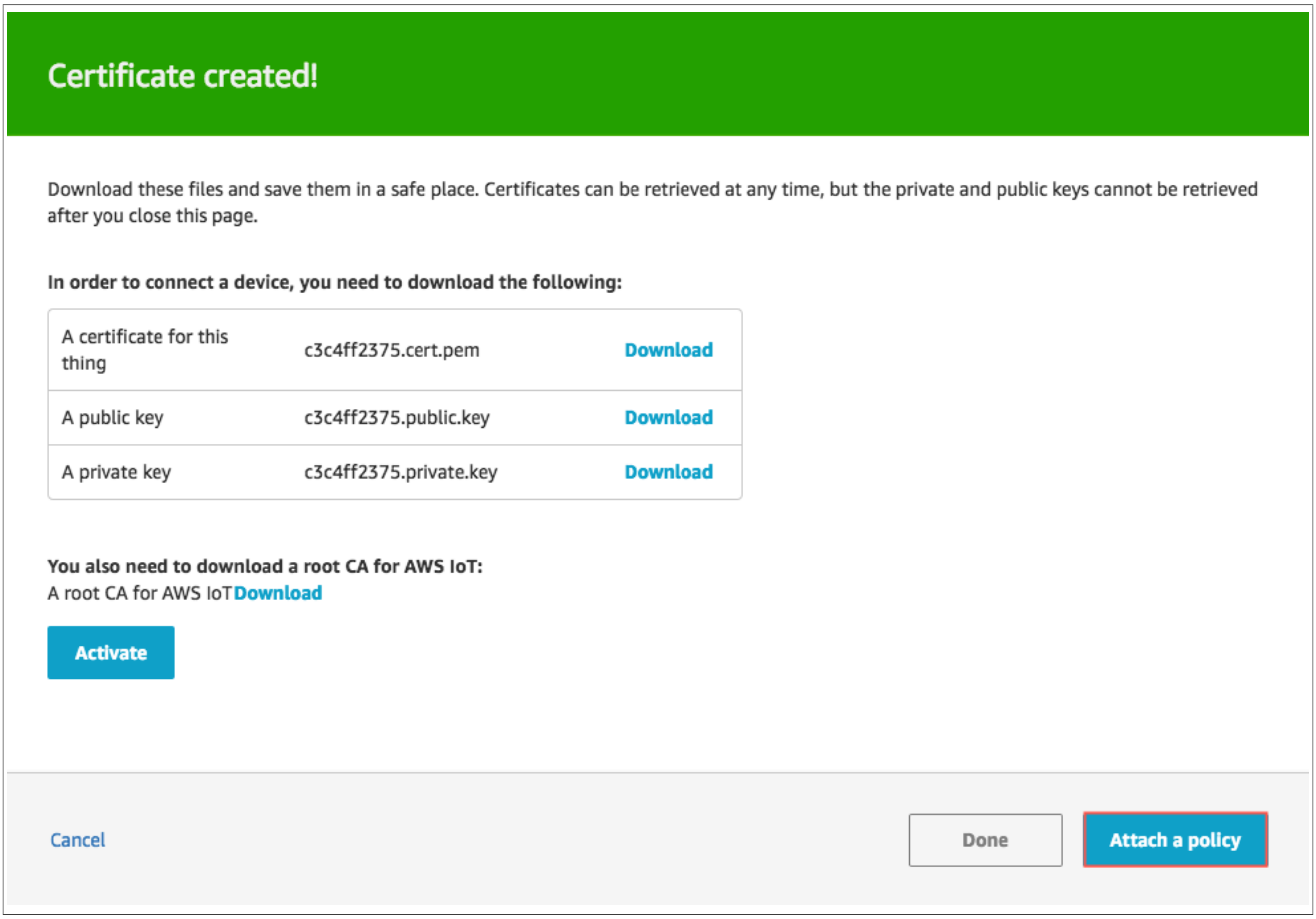
Use my certificate

Register your CA certificate and use your own certificates for one or many devices.

Skip certificate and create thing

You will need to add a certificate to your thing later before your device can connect to AWS IoT.

7. On the **Certificate created!** page, download your public and private keys, certificate, and root certificate authority (CA).
8. Choose **Download** for your certificate.
9. Choose **Download** for your private key.
10. Choose **Download** for the Amazon root CA.
    A new webpage is displayed.
11. Choose **RSA 2048-bit key: Amazon Root CA 1**.
    This opens another webpage with the text of the root CA certificate.
12. Copy this text and paste it into a file named `Amazon_Root_CA_1.pem`.
    Most web browsers save downloaded files into a Downloads directory. You copy these files to a different directory when you run the sample applications. Choose **Activate** to activate the X.509 certificate, and then choose **Attach a policy**.



Certificate created!

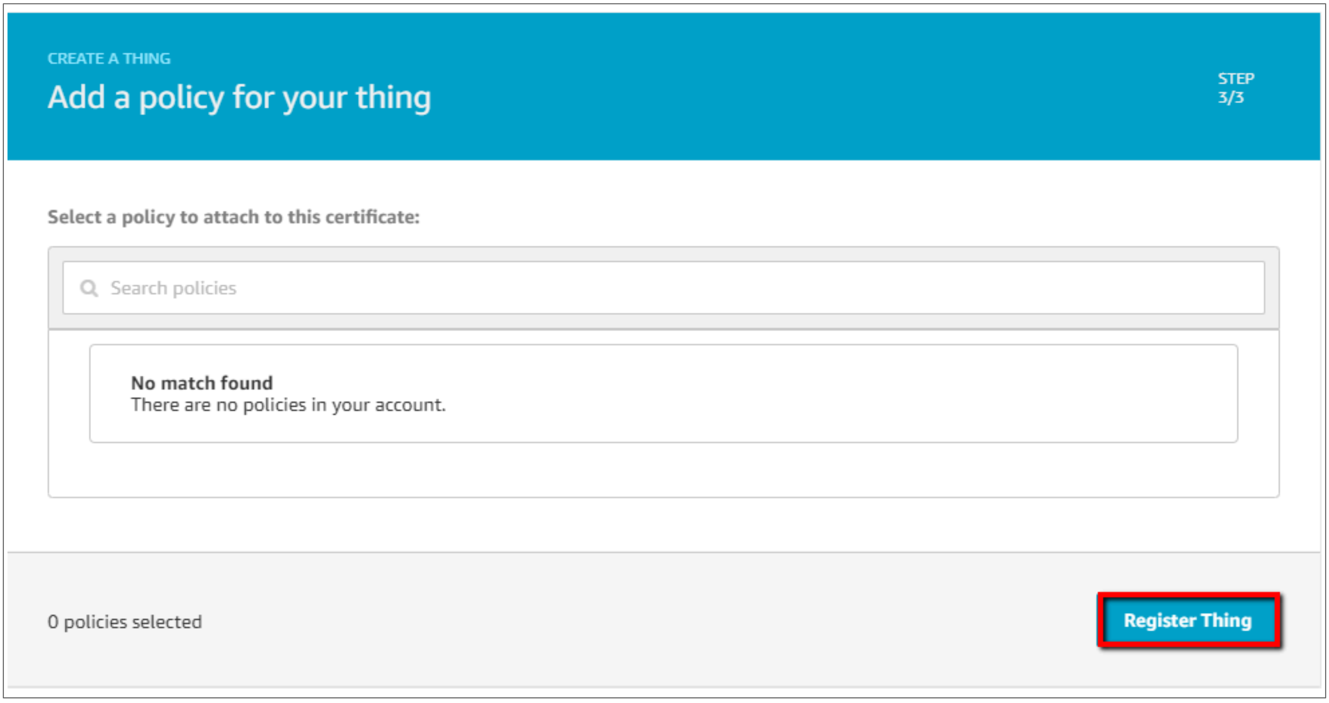
Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

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

A certificate for this thing	c3c4ff2375.cert.pem	Download
A public key	c3c4ff2375.public.key	Download
A private key	c3c4ff2375.private.key	Download

You also need to download a root CA for AWS IoT:
A root CA for AWS IoT **Download**

13. On the **Add a policy for your thing** page, choose **Register Thing**. After you register your thing, create and attach a new policy to the certificate.



CREATE A THING

Add a policy for your thing

STEP 3/3

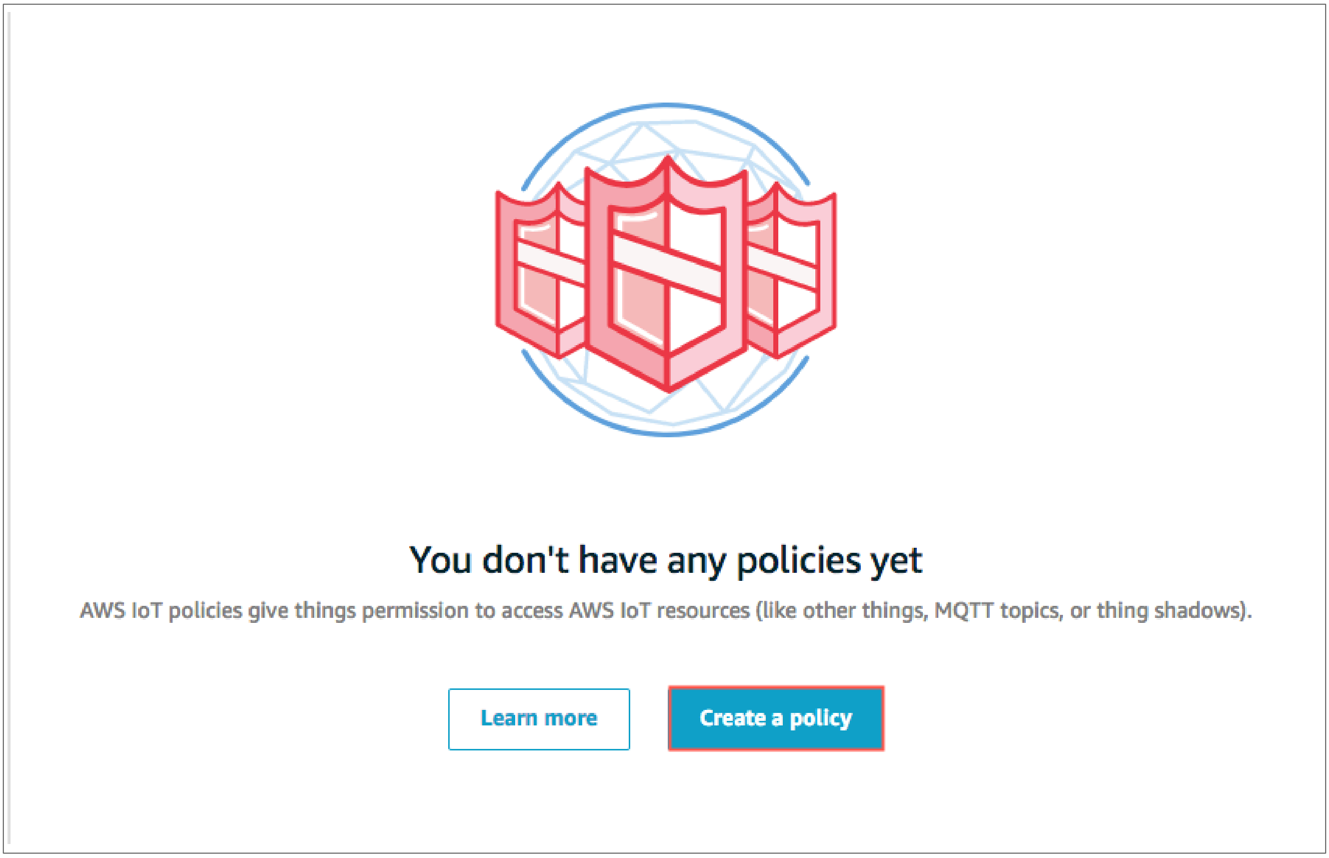
Select a policy to attach to this certificate:

No match found
There are no policies in your account.

0 policies selected

4.2 Create an AWS IoT policy

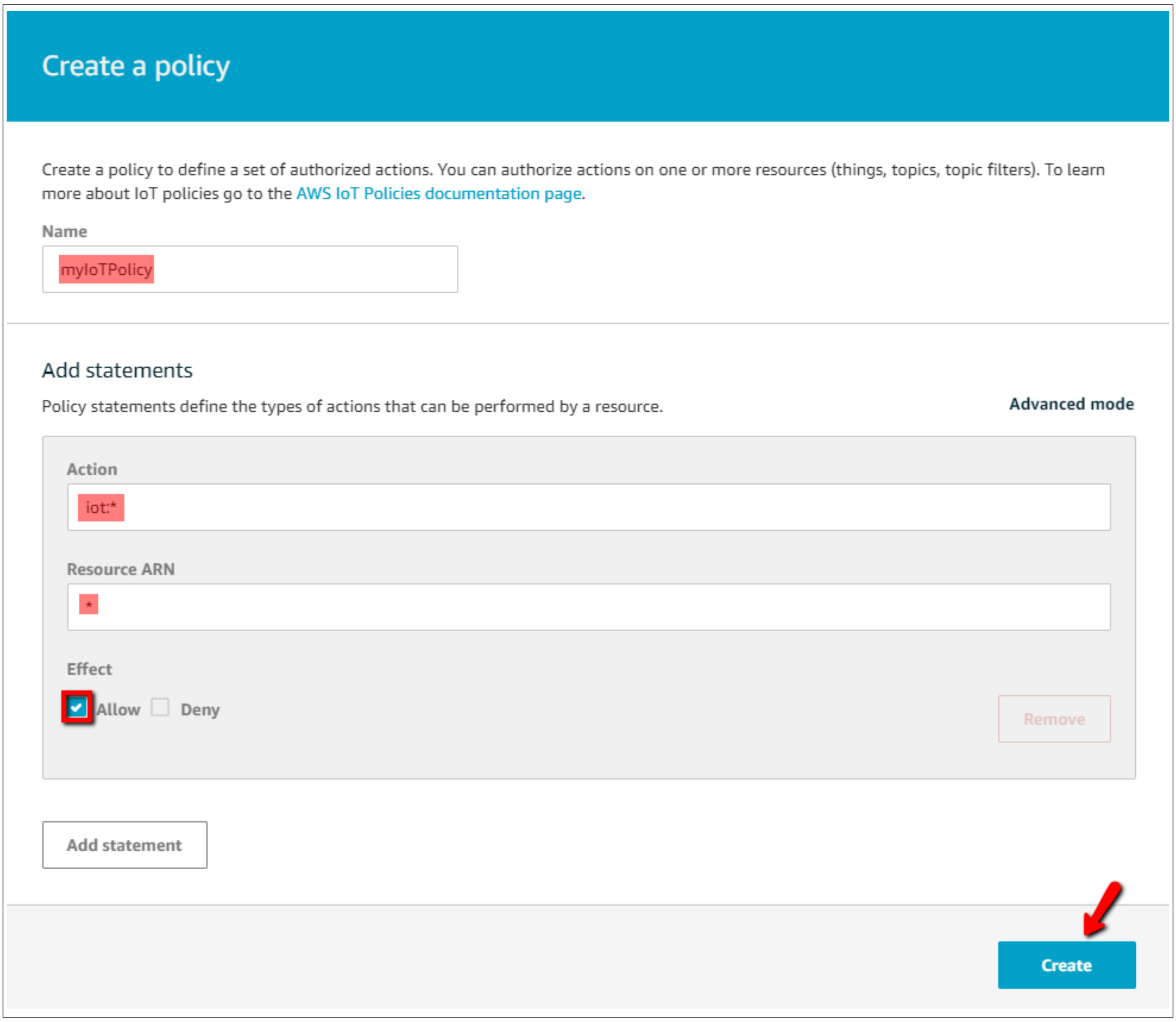
1. In the left navigation pane, choose **Secure**, and then choose **Policies**. On the **You don't have a policy yet** page, choose **Create a policy**.



You don't have any policies yet

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

2. On the **Create a policy** page, in the **Name** field, enter a name for the policy (for example, MyIotPolicy).
3. In the **Action** field, enter **iot:Connect**.
4. In the **Resource ARN** field, enter *****.
5. Select the **Allow** checkbox.
   This allows all clients to connect to AWS IoT.
6. After you have entered the information for your policy, choose **Create**.



Create a policy

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

Name

myIoTPolicy

Add statements

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

Action

iot:*

Resource ARN

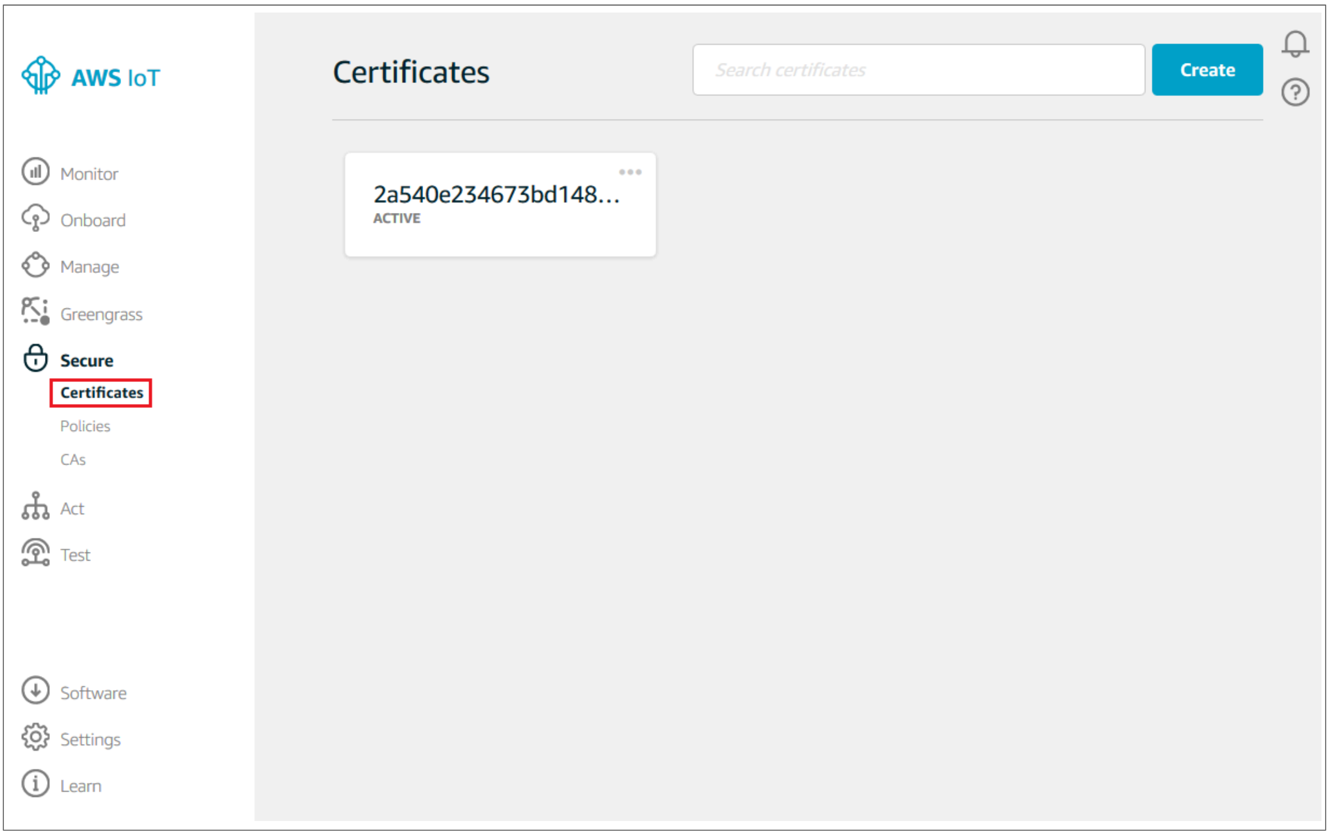
*

Effect

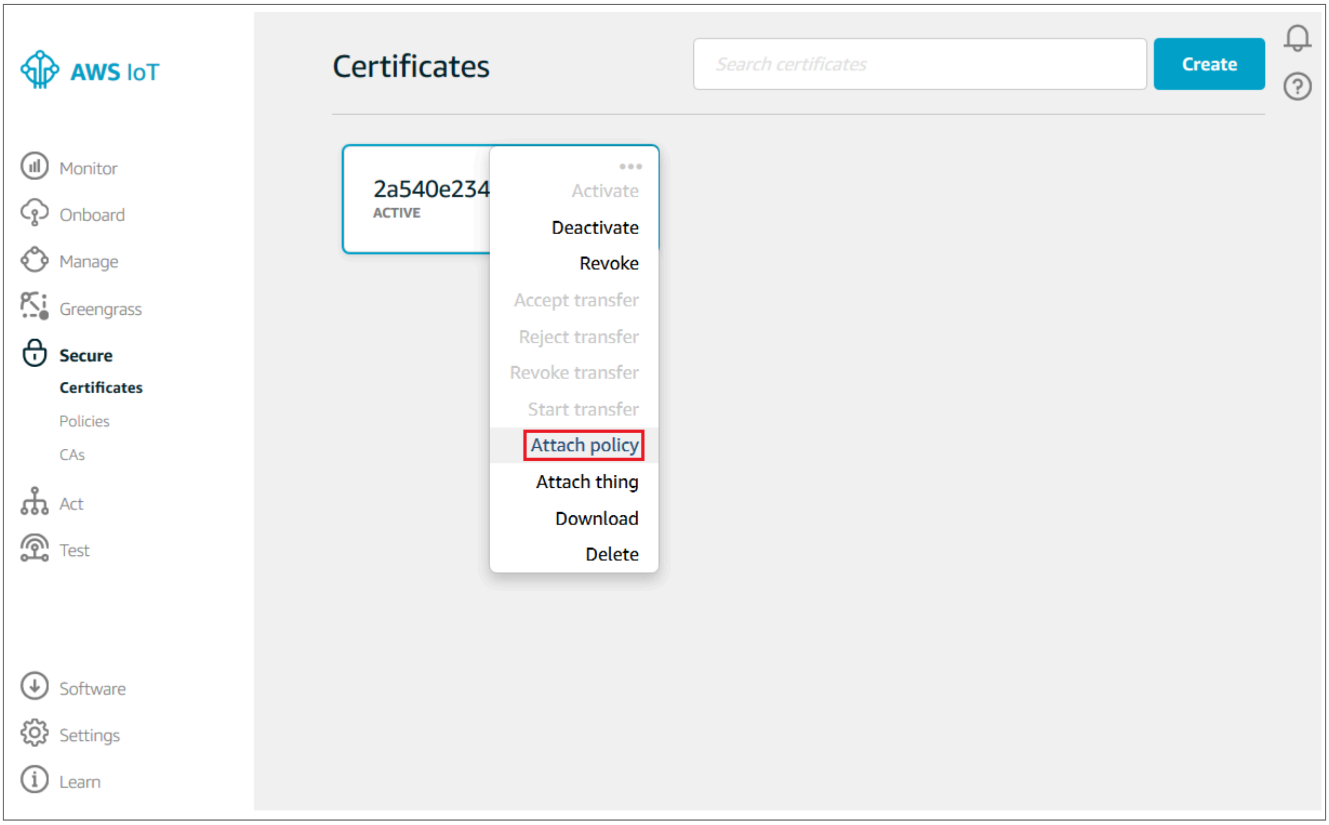
☑ Allow □ Deny

4.3 Attach an AWS IoT policy to a device certificate

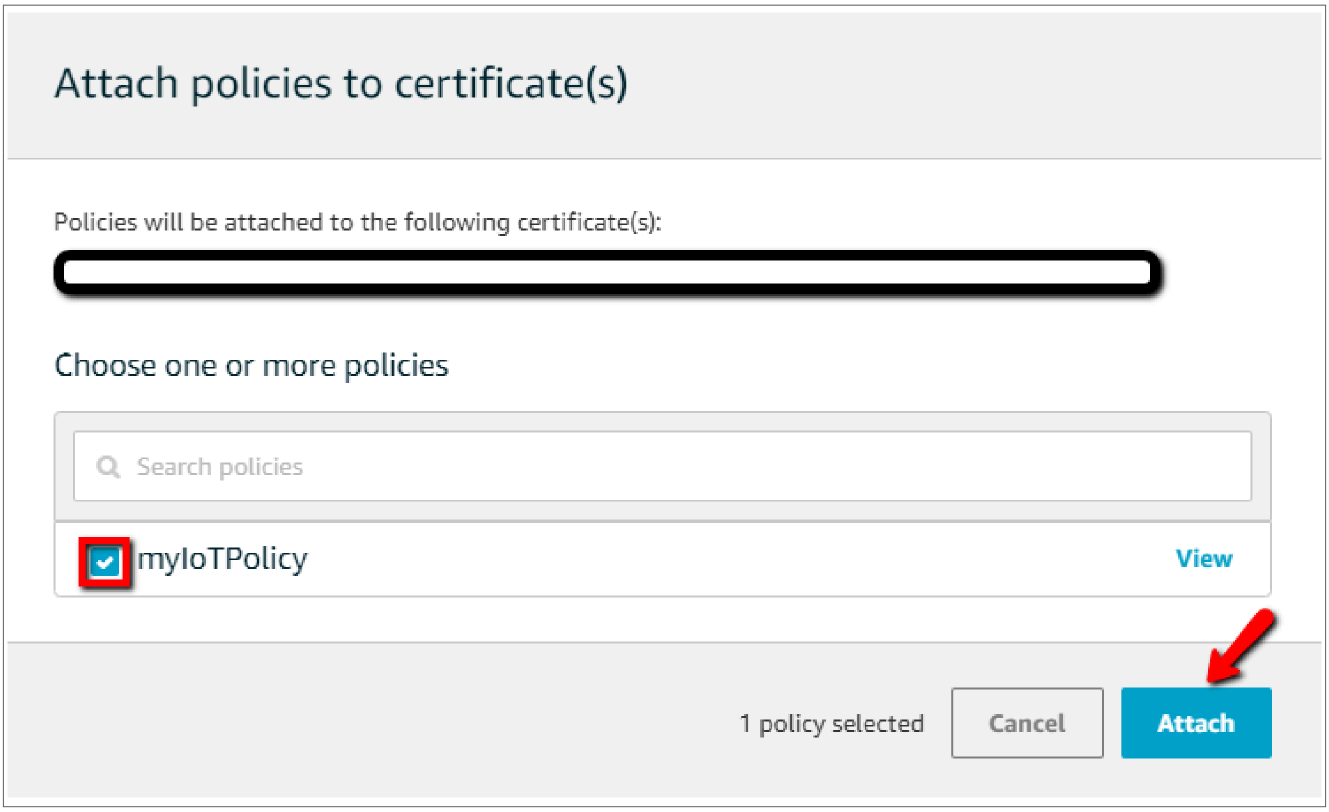
1. In the left navigation pane, choose **Secure**, and then choose **Certificates**.



2. In the box for the certificate you created, choose **...** to open a drop-down menu, and then choose **Attach policy**.



3. In **Attach policies to certificate(s)**, select the checkbox next to the policy you created in the previous step, and then choose **Attach**.



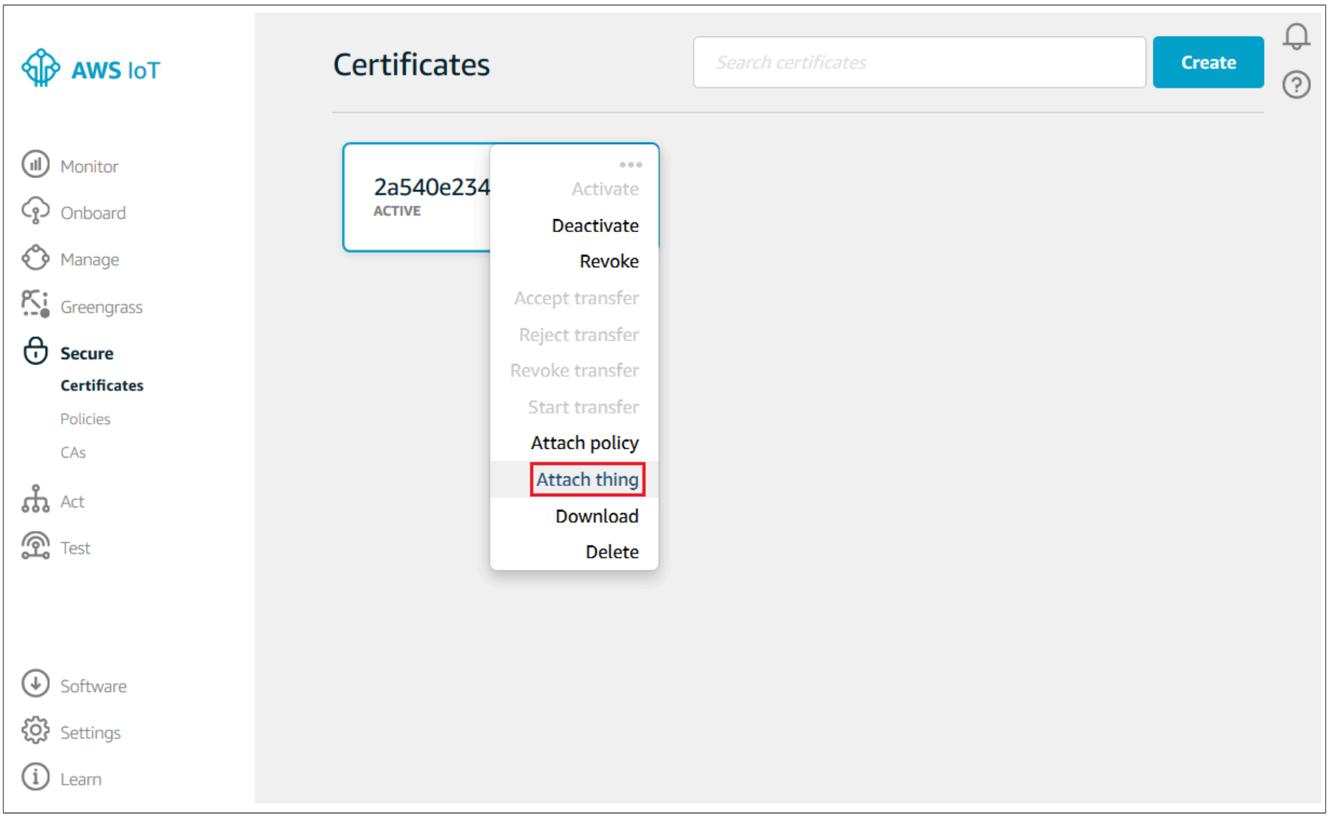
Attach policies to certificate(s)

Policies will be attached to the following certificate(s):

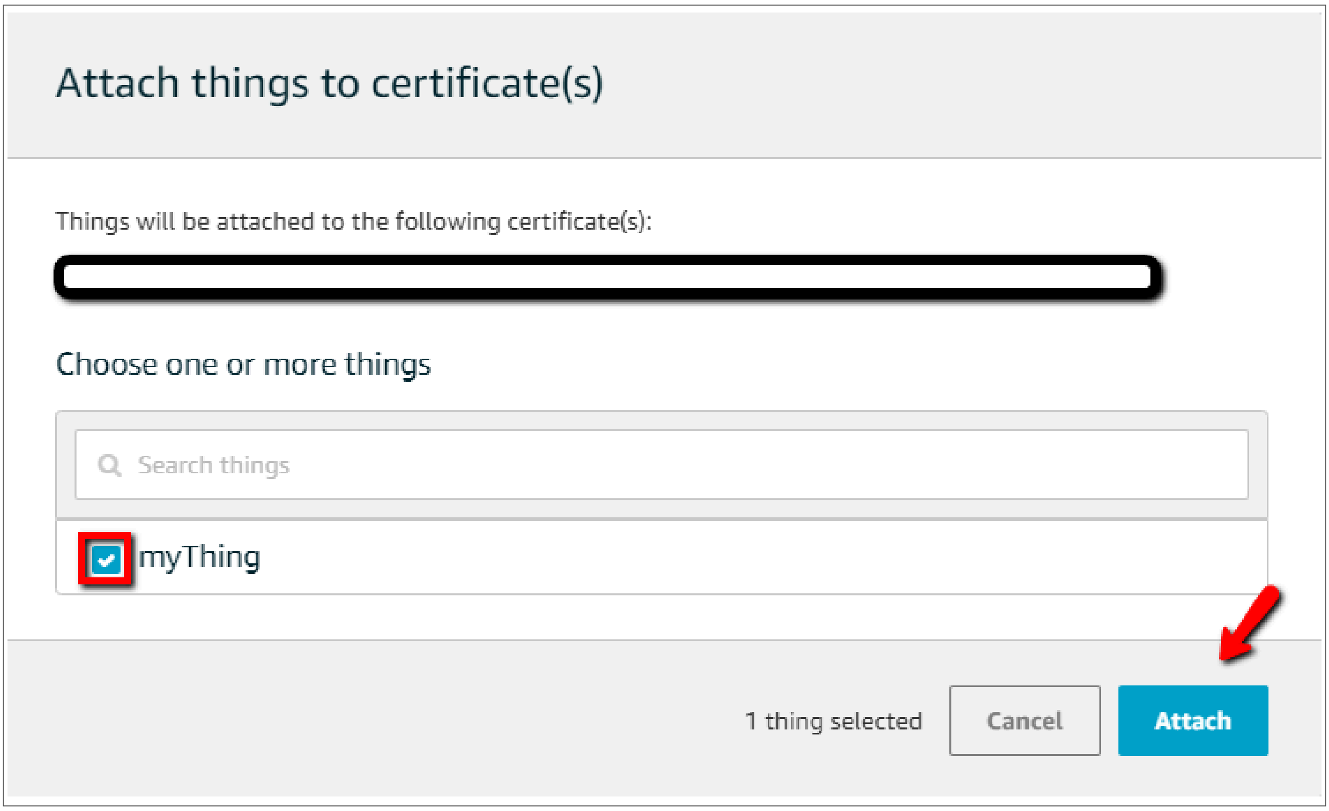
Choose one or more policies

4.4 Attach a certificate to a thing

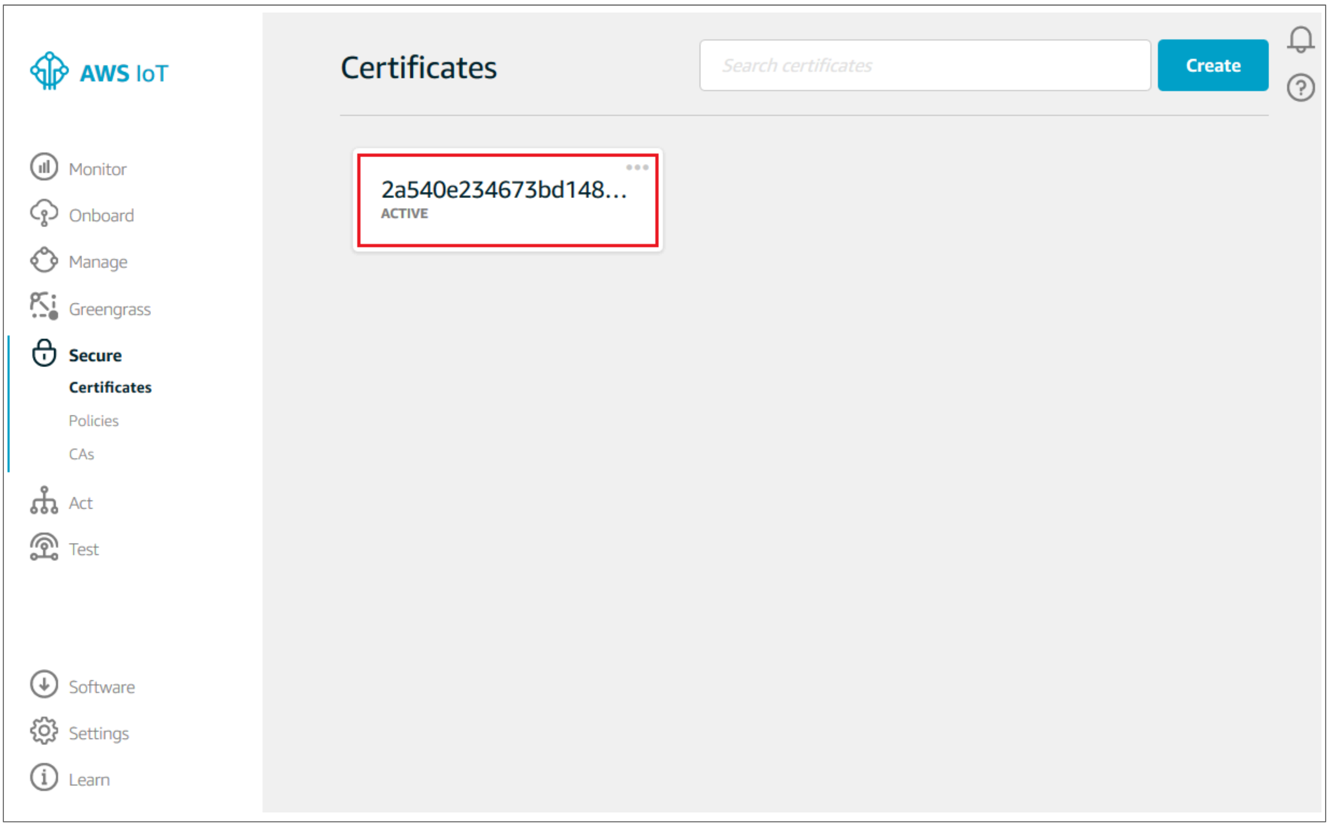
1. In the box for the certificate you created, choose **...** to open a drop-down menu, and then choose **Attach thing**.



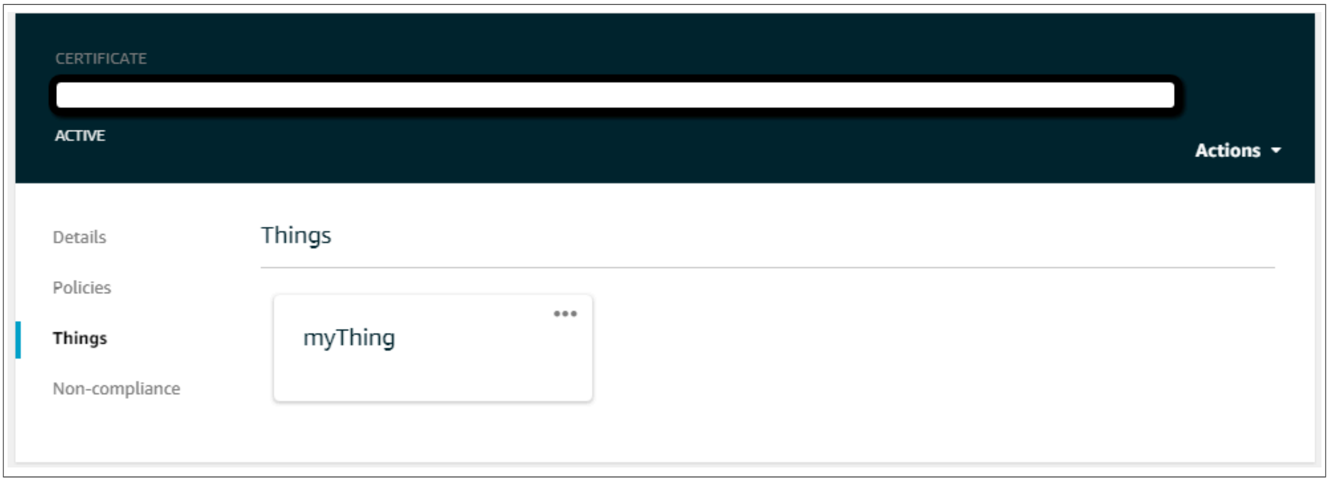
2. In **Attach things to certificate(s)**, select the checkbox next to the thing you registered, and then choose **Attach**.


Attach things to certificate(s)
Things will be attached to the following certificate(s):
[redacted]
Choose one or more things
Search things
☑ myThing
1 thing selected
Cancel
Attach

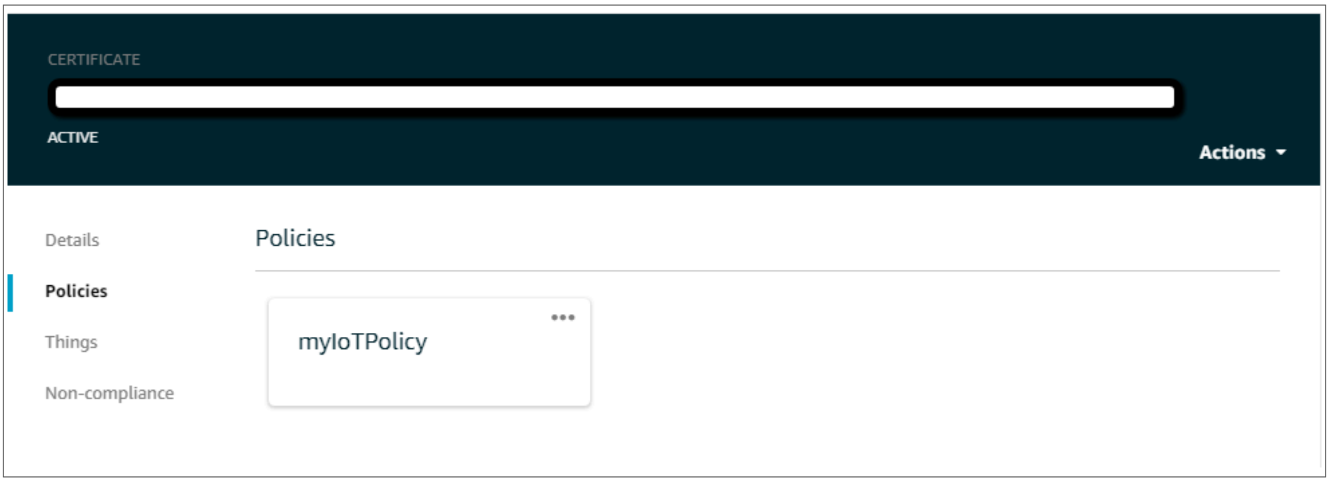
3. To verify that the thing is attached, select the box for the certificate.



4. On the **Details** page for the certificate, in the left navigation pane, choose **Things**.



5. To verify that the policy is attached, on the **Details** page for the certificate, in the left navigation pane, choose **Policies**.

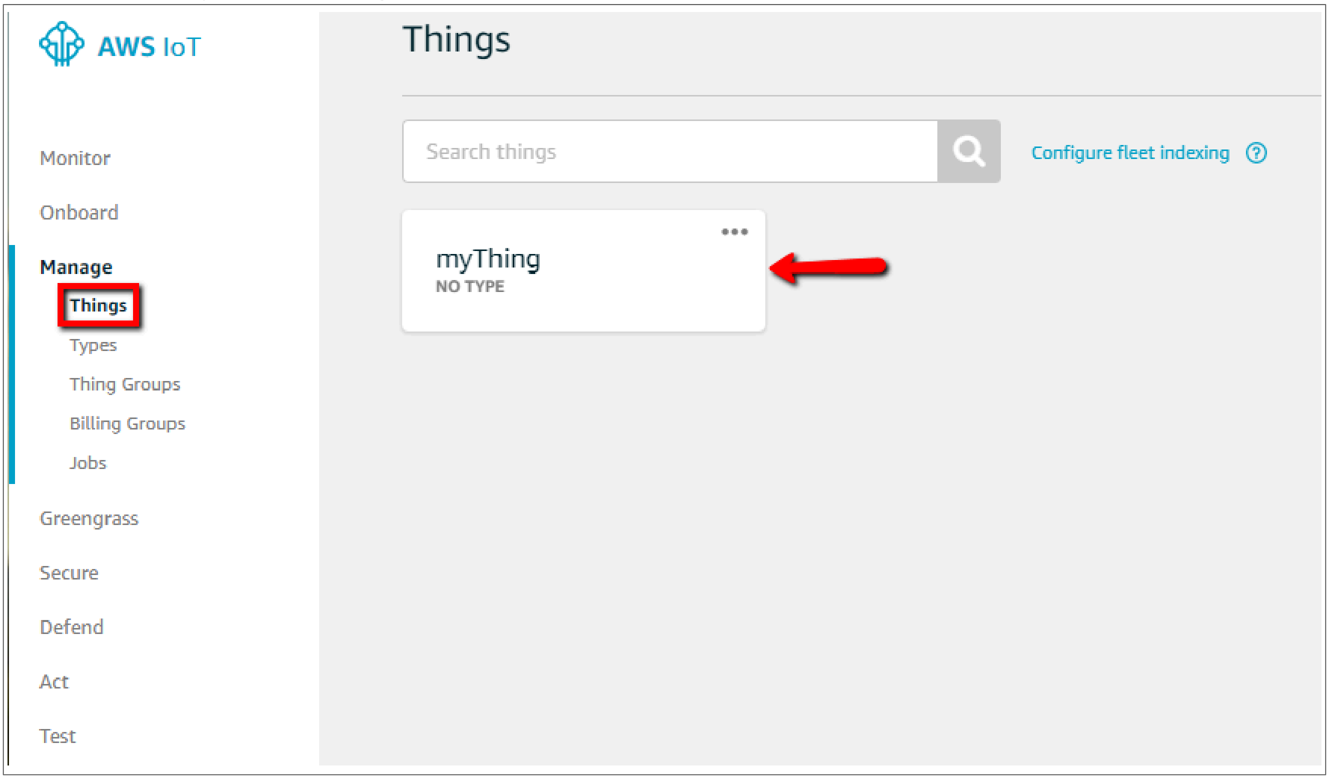


5 Configure the device

The related SDK code folder is available here: `SDK_2.x.x_EVK-MIMXRT1060\boards\evkmimxrt1060\aws_examples\ota_demo_enet`.

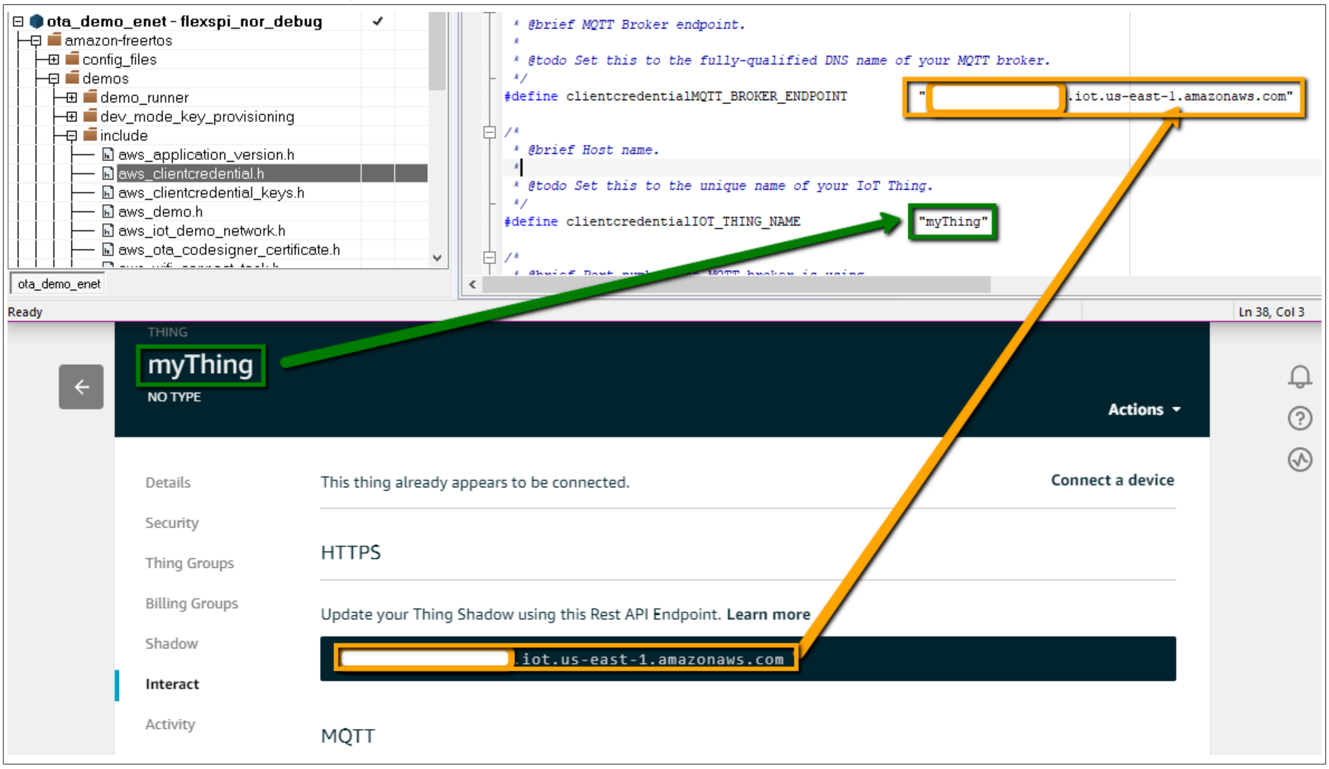
5.1 aws_clientcredential.h

1. Open the AWS IoT console website https://console.aws.amazon.com/iot/
2. On the **Welcome to the AWS IoT Console** page, in the navigation pane, choose **Manage – Things** select the previously created **Thing**.



3. In the navigation pane, choose **Interact,** copy the **REST API** endpoint and **IoT Thing** name.

4. Inside the OTA project, open amazon-freertos – demos – include – aws_clientcredential.h and set the **REST_API** and **IoT Thing** name obtained in the previous step.



ota_demo_enet - flexspi_nor_debug
amazon-freertos
config_files
demos
demo_runner
dev_mode_key_provisioning
include
aws_application_version.h
aws_clientcredential.h
aws_clientcredential_keys.h
aws_demo.h
aws_iot_demo_network.h
aws_ota_codesigner_certificate.h
ota_demo_enet
Ready
Ln 38, Col 3

```
 * @brief MQTT Broker endpoint.
 *
 * @todo Set this to the fully-qualified DNS name of your MQTT broker.
 */
#define clientcredentialMQTT_BROKER_ENDPOINT     "[redacted].iot.us-east-1.amazonaws.com"

/*
 * @brief Host name.
 *
 * @todo Set this to the unique name of your IoT Thing.
 */
#define clientcredentialIOT_THING_NAME     "myThing"
```

This thing already appears to be connected.

HTTPS

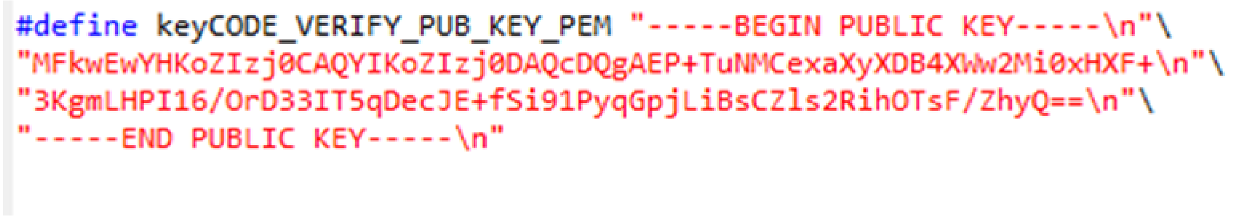
Update your Thing Shadow using this Rest API Endpoint. **Learn more**

[redacted].iot.us-east-1.amazonaws.com

MQTT

5.2 aws_clientcredential_keys.h

1. Open file with certificate as mentioned in Section 4.1 "Create an AWS IoT thing", step 8, using a text editor.
2. Copy all the content, paste the information in the: `keyCLIENT_CERTIFICATE_PEM`.
   Note: *Ensure to add " at the beginning of a line and* **\n"\** *on every line break.*
3. In same way update `keyCLIENT_PRIVATE_KEY_PEM` with content of private key file. See, Section 4.1 "Create an AWS IoT thing", step 9.
4. In same way update `keyCODE_VERIFY_PUB_KEY_PEM` with content of code signing public key file See, Section 2.5 "Creating a code-signing certificate", step 5.

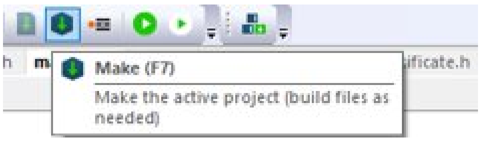


```
#define keyCODE_VERIFY_PUB_KEY_PEM "-----BEGIN PUBLIC KEY-----\n"\
"MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEP+TuNMCexaXyXDB4XWw2Mi0xHXF+\n"\
"3KgmLHPI16/OrD33IT5qDecJE+fSi91PyqGpjLiBsCZls2RihOTsF/ZhyQ==\n"\
"-----END PUBLIC KEY-----\n"
```

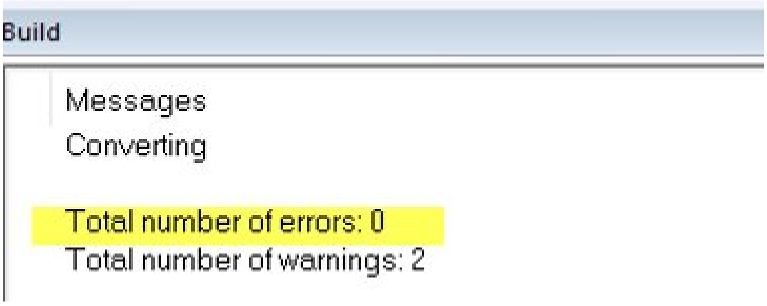
Figure 1. Example key

5.3 Build

1. Click the make button to start building the application.



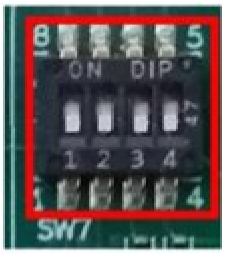
2. If the build is successful, Zero errors message is printed in build console.



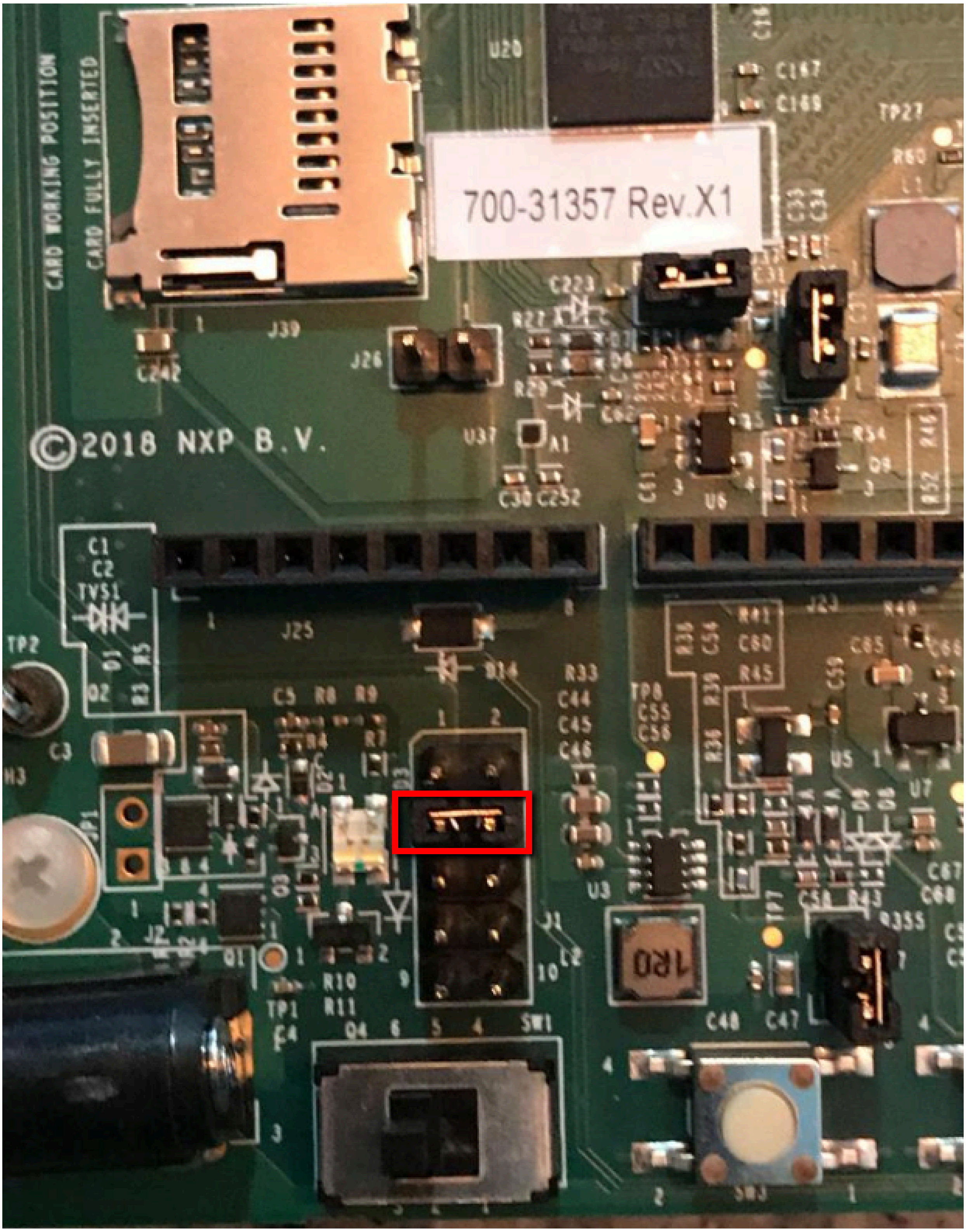
```
Messages
Converting

Total number of errors: 0
Total number of warnings: 2
```

5.4 Programming mcu-boot into flash

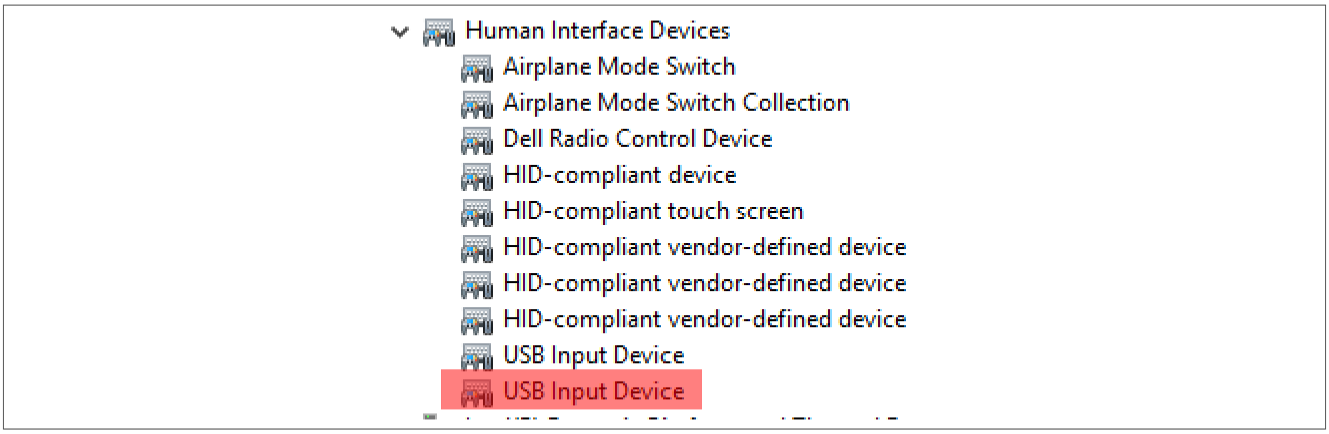
1. Set all SW7 positions to off.



2. Locate J1, then move the jumper to **3-4**.

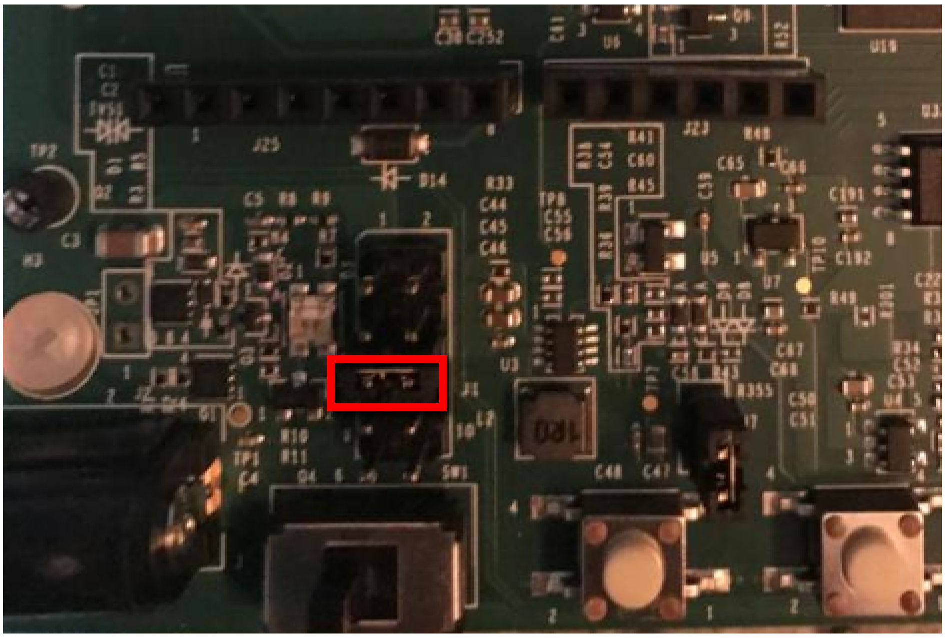


3. Connect the board to PC via J9 USB connector.
4. Reset the board using **SW3**, then make sure that your RT1060-EVK gets enumerated like Human Interface Devices – USB Input device.

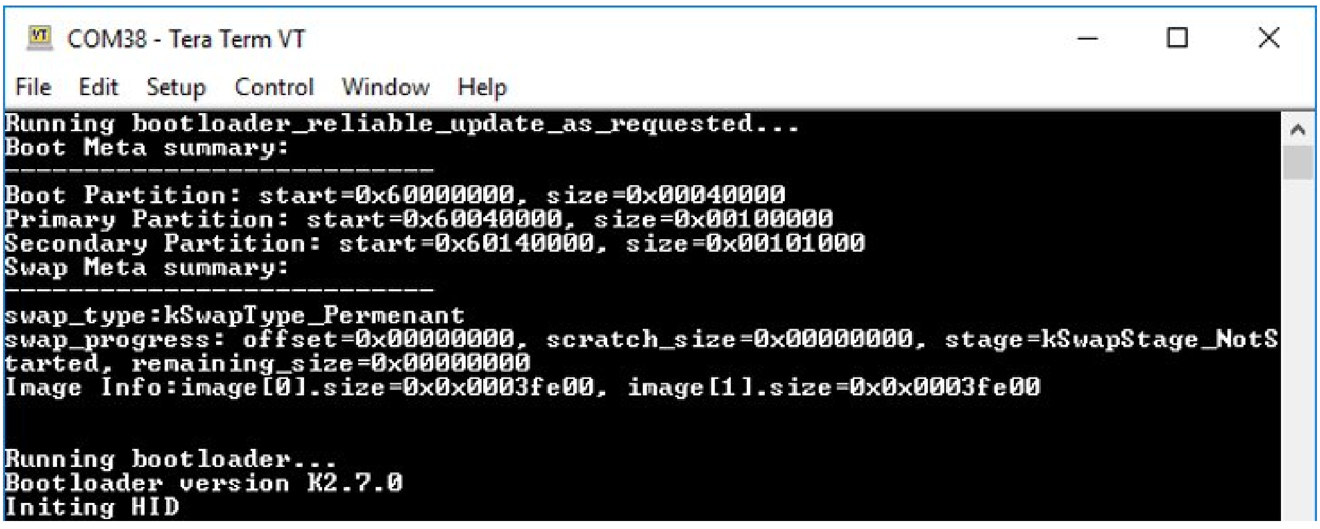


Human Interface Devices
Airplane Mode Switch
Airplane Mode Switch Collection
Dell Radio Control Device
HID-compliant device
HID-compliant touch screen
HID-compliant vendor-defined device
HID-compliant vendor-defined device
HID-compliant vendor-defined device
USB Input Device
USB Input Device

5. Open the Command Prompt window.
6. Execute the following commands. It is recommended, but not required, to have bash interpreter at hand. Git bash does the job https://gitforwindows.org/ > cd ..\OTA_Bootloader_Scripts-4e081f\OTA_Bootloader_Scripts_0.5 > generate_ota_bootloader_and_program_it_to_flash.sh.
7. Disconnect the USB cable from the J9 USB connector.
8. Set SWD7[1:4]:0010.
9. Return J1 jumper to the default setting **5-6**.



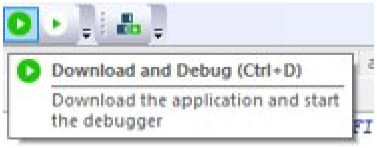
10. Connect the RT1060-EVK to the PC using the OpenSDA USB connector J41, mimxrt1060-evk, and SCH rev A2. Use some terminal application to connect the virtual com port to see the console.
11. Reset the EVK using SW3 at this moment that you should be able to see bootloader messages being printed on a terminal.



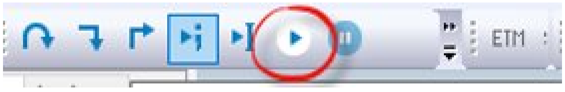
```
Running bootloader_reliable_update_as_requested...
Boot Meta summary:
---------------------------
Boot Partition: start=0x60000000, size=0x00040000
Primary Partition: start=0x60040000, size=0x00100000
Secondary Partition: start=0x60140000, size=0x00101000
Swap Meta summary:
---------------------------
swap_type:kSwapType_Permenant
swap_progress: offset=0x00000000, scratch_size=0x00000000, stage=kSwapStage_NotS
tarted, remaining_size=0x00000000
Image Info:image[0].size=0x0x0003fe00, image[1].size=0x0x0003fe00


Running bootloader...
Bootloader version K2.7.0
Initing HID
```

5.5 Flashing the OTA Agent application

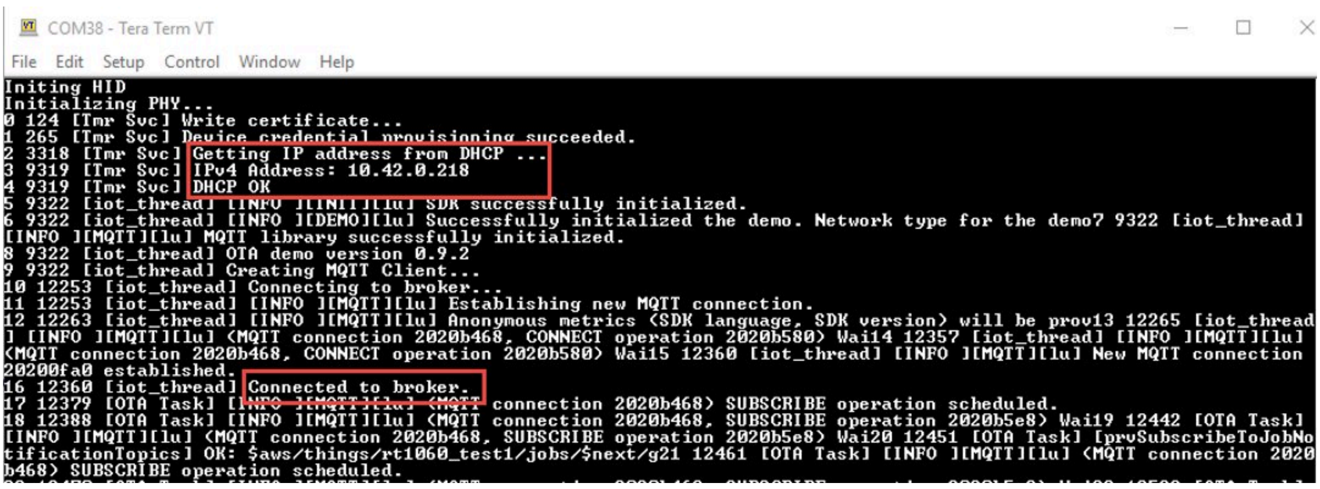
1. Attach the Ethernet cable with Internet connection and local DHCP server.
2. Click the download and debug button to start flashing the device.


Download and Debug (Ctrl+D)
Download the application and start the debugger

3. When the device is flashed, the debug pointer turns green the main entry point.
4. Click the Go button to start running the program.



Double check that there are MQTT AWS messages on the terminal.



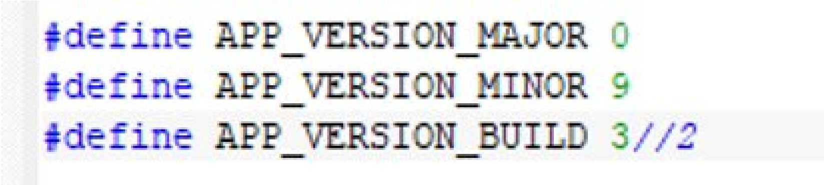
```
Initing HID
Initializing PHY...
0 124 [Tmr Svc] Write certificate...
1 265 [Tmr Svc] Device credential provisioning succeeded.
2 3318 [Tmr Svc] Getting IP address from DHCP ...
3 9319 [Tmr Svc] IPv4 Address: 10.42.0.218
4 9319 [Tmr Svc] DHCP OK
5 9322 [iot_thread] [INFO ][INIT][lu] SDK successfully initialized.
6 9322 [iot_thread] [INFO ][DEMO][lu] Successfully initialized the demo. Network type for the demo7 9322 [iot_thread]
[INFO ][MQTT][lu] MQTT library successfully initialized.
8 9322 [iot_thread] OTA demo version 0.9.2
9 9322 [iot_thread] Creating MQTT Client...
10 12253 [iot_thread] Connecting to broker...
11 12253 [iot_thread] [INFO ][MQTT][lu] Establishing new MQTT connection.
12 12263 [iot_thread] [INFO ][MQTT][lu] Anonymous metrics (SDK language, SDK version) will be prov13 12265 [iot_thread
] [INFO ][MQTT][lu] (MQTT connection 2020b468, CONNECT operation 2020b580) Wai14 12357 [iot_thread] [INFO ][MQTT][lu]
(MQTT connection 2020b468, CONNECT operation 2020b580) Wai15 12360 [iot_thread] [INFO ][MQTT][lu] New MQTT connection
20200fa0 established.
16 12360 [iot_thread] Connected to broker.
17 12379 [OTA Task] [INFO ][MQTT][lu] (MQTT connection 2020b468) SUBSCRIBE operation scheduled.
18 12388 [OTA Task] [INFO ][MQTT][lu] (MQTT connection 2020b468, SUBSCRIBE operation 2020b5e8) Wai19 12442 [OTA Task]
[INFO ][MQTT][lu] (MQTT connection 2020b468, SUBSCRIBE operation 2020b5e8) Wai20 12451 [OTA Task] [prvSubscribeToJobNo
tificationTopics] OK: $aws/things/rt1060_test1/jobs/$next/g21 12461 [OTA Task] [INFO ][MQTT][lu] (MQTT connection 2020
b468) SUBSCRIBE operation scheduled.
```

5. Stop the debug session.

6 OTA

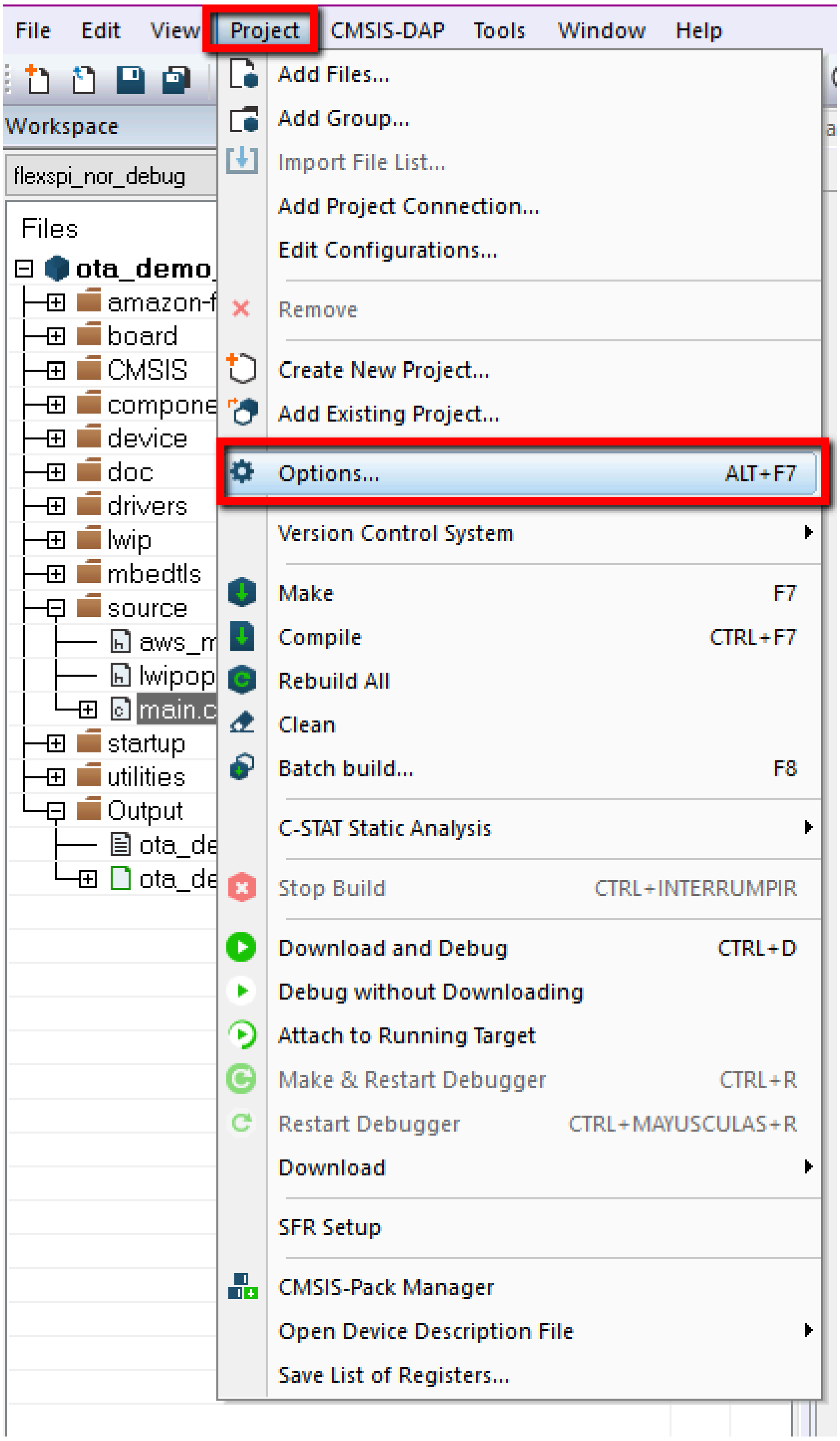
6.1 Create new image

1. Open `ota_config.h`.
2. Change any of the APP_VERSION macros to a higher number.

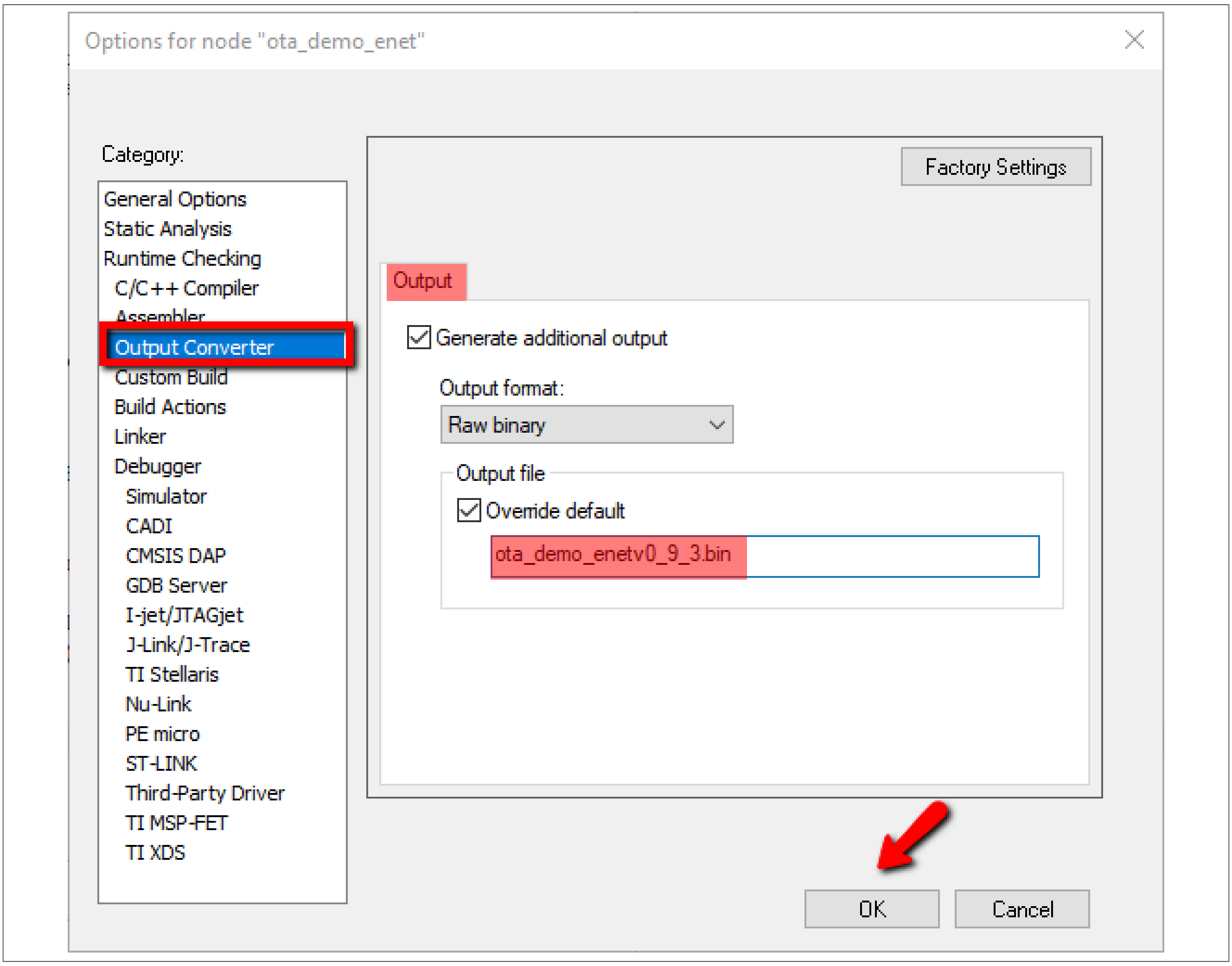


```
#define APP_VERSION_MAJOR 0
#define APP_VERSION_MINOR 9
#define APP_VERSION_BUILD 3//2
```

3. Open **Project > Options**.



4. In the Category section, choose **Output Converter**.
5. Change the name of the binary so it matches the version change then click **OK**.



Options for node "ota_demo_enet"

Output

☑ Generate additional output

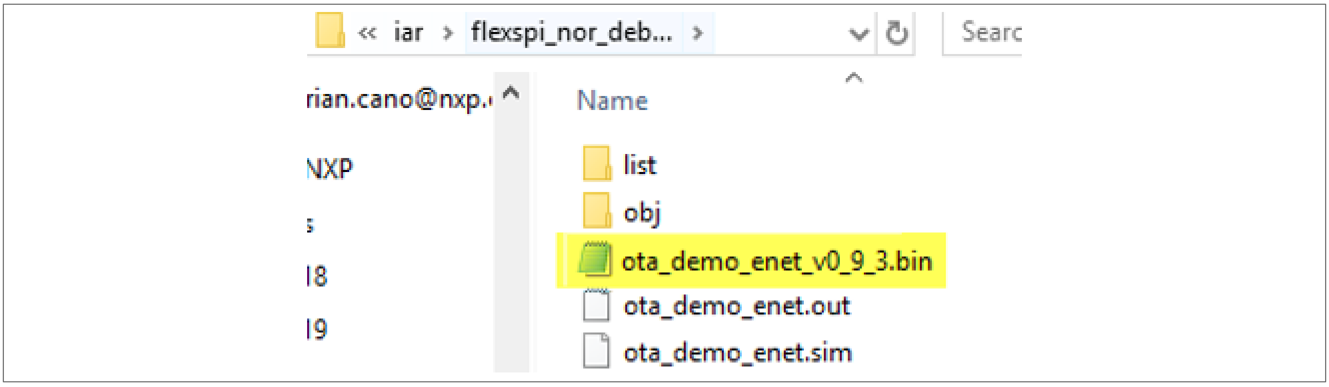
Output format: Raw binary

Output file

☑ Override default

ota_demo_enetv0_9_3.bin

6. Use the make button to build and generate the binary. Look for the binary inside the …boards \evkmimxrt1060\aws_examples\ota_demo_enet\iar\flexspi_nor_debug folder.



« iar › flexspi_nor_deb... ›

Name

list

obj

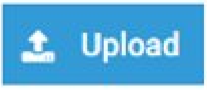
ota_demo_enet_v0_9_3.bin

ota_demo_enet.out

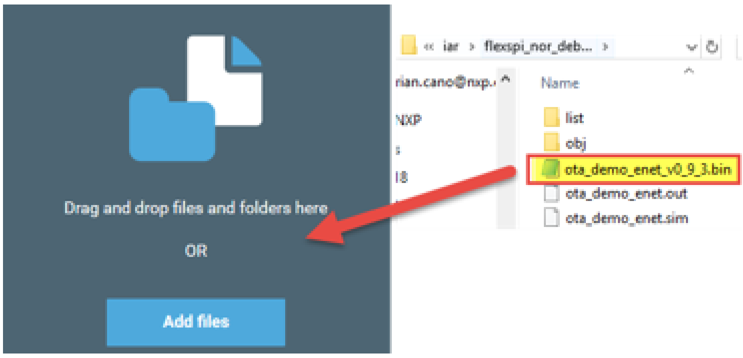
ota_demo_enet.sim

6.2 Uploading the binary to the S3 bucket

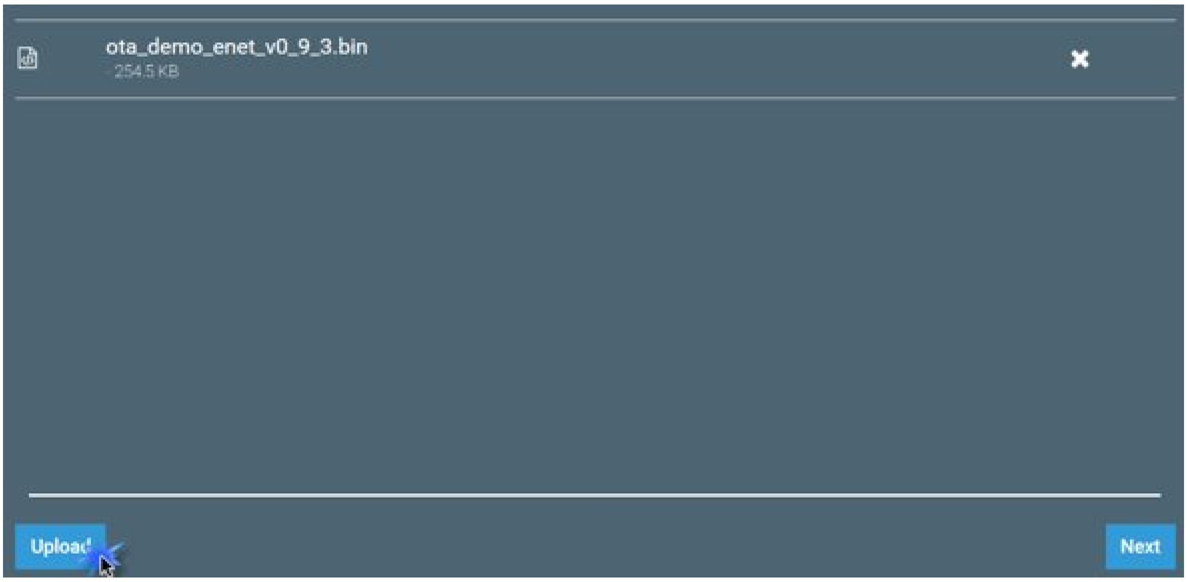
1. Use AWS console to open the S3 service https://console.aws.amazon.com/s3.
2. Select the previously created bucket.
3. Click **Upload**.



4. Drag and drop the ota_demo_enet_v0_9_3.bin binary.

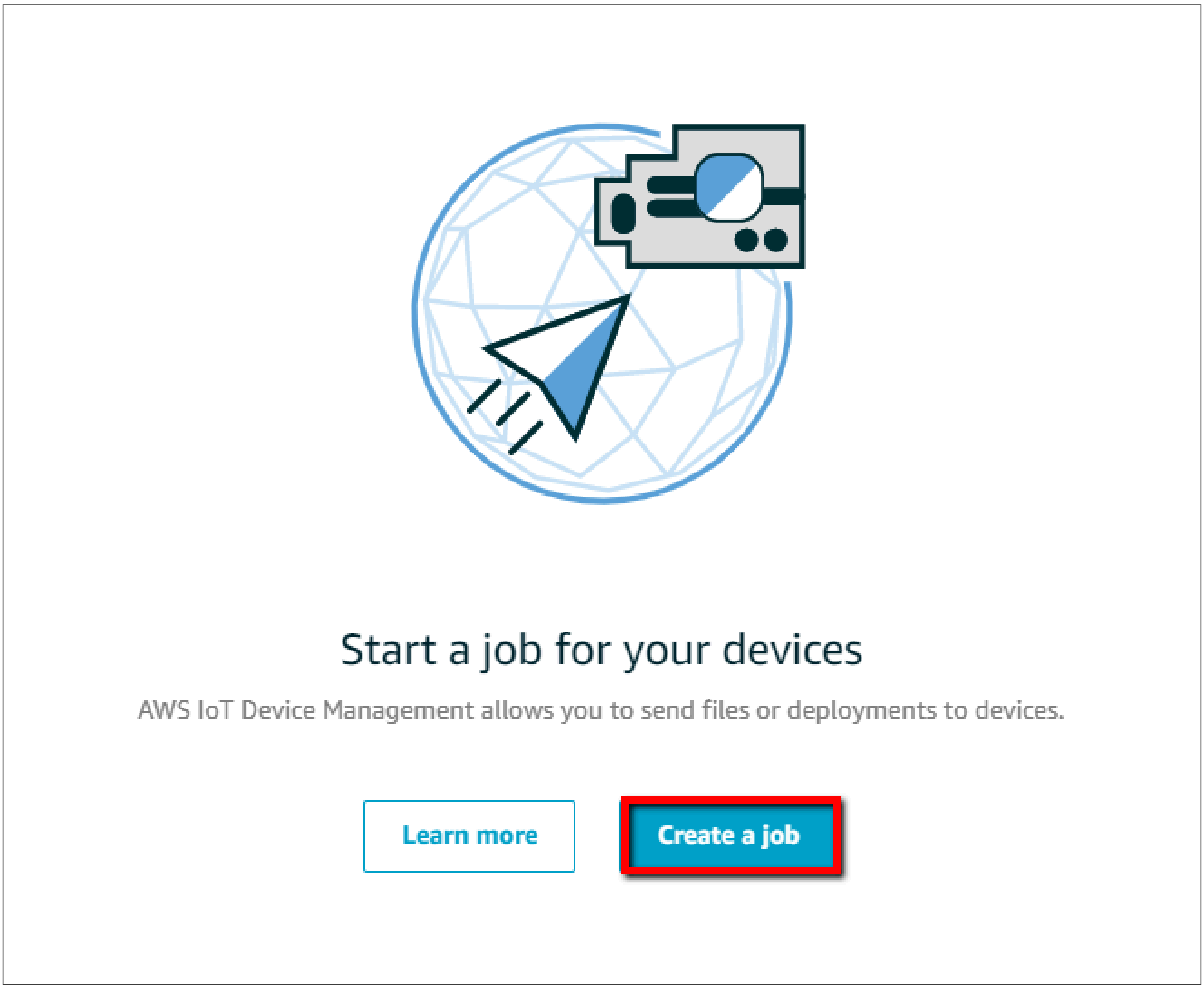


5. Click **Upload**.



6.3 Create OTA Job

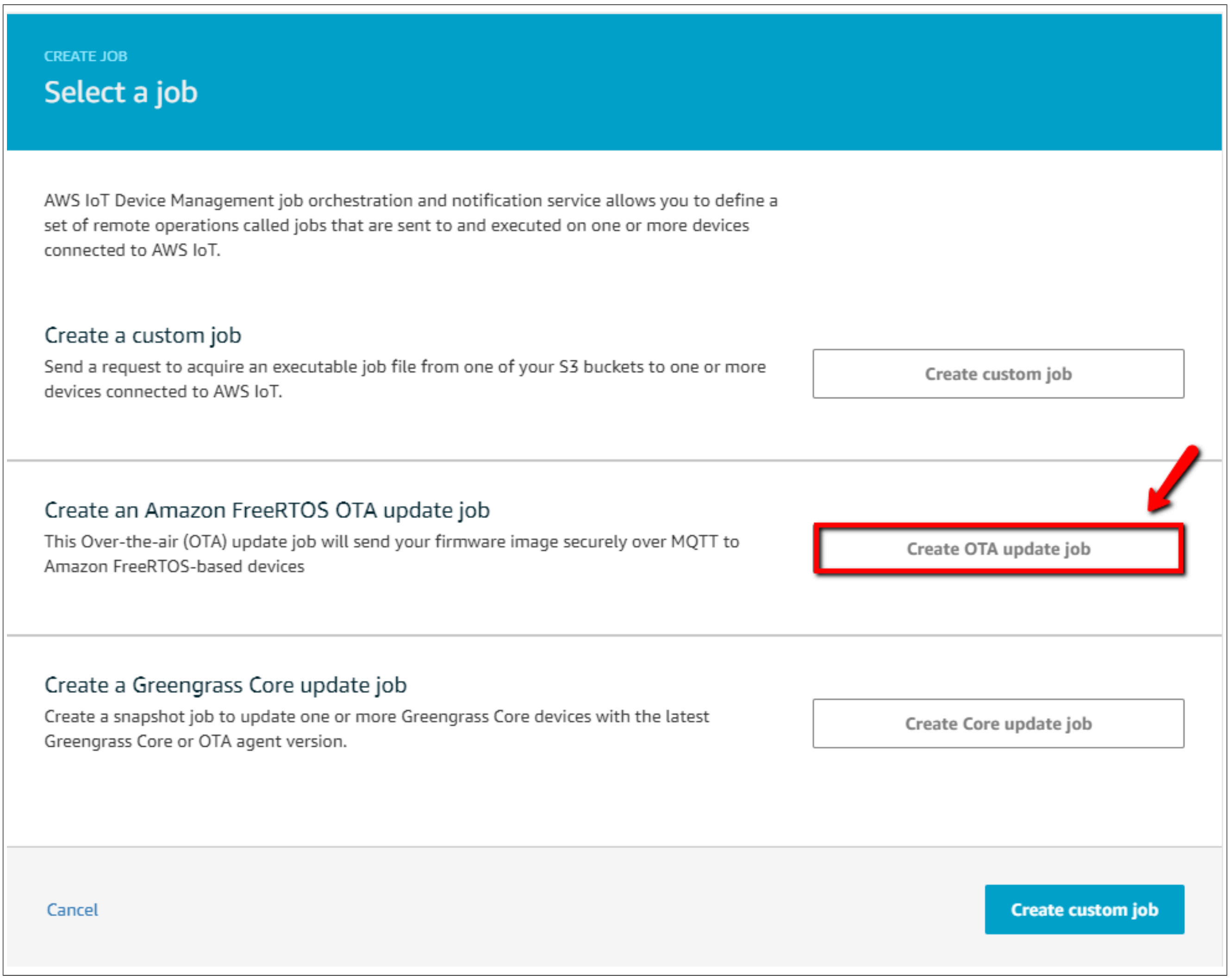
1. Open the AWS IoT console website https://console.aws.amazon.com/iot/.
2. On the **Welcome to the AWS IoT Console** page, in the navigation pane, choose **Manage – Jobs**.
3. Select **Create**.



Start a job for your devices

AWS IoT Device Management allows you to send files or deployments to devices.

4. Under **Create an Amazon FreeRTOS Over-the-Air (OTA) update job**, choose **Create OTA update job**.



CREATE JOB

Select a job

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.

Create a custom 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.

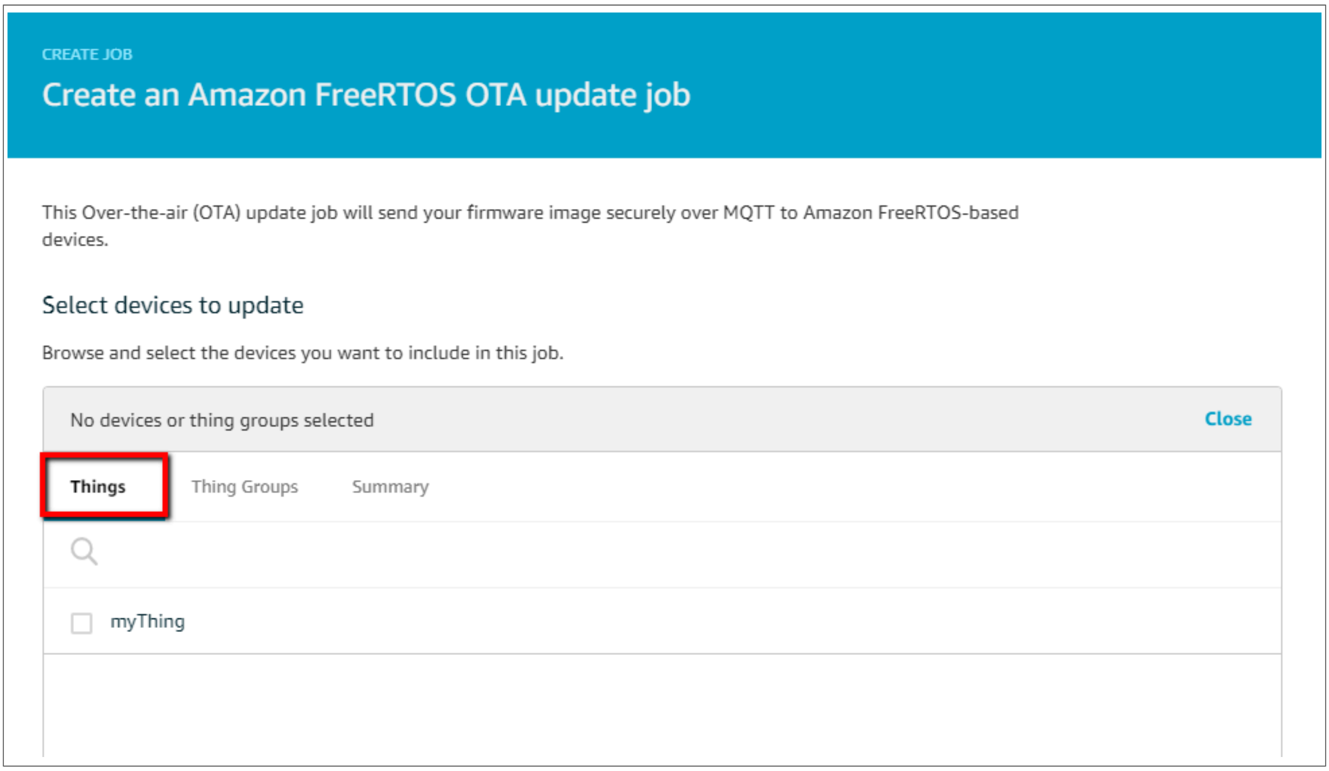
Create an Amazon FreeRTOS OTA update job

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

Create a Greengrass Core update job

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

5. Under **Select devices to update**, choose **Select**. To update a single device, choose the **Things** tab.



CREATE JOB

Create an Amazon FreeRTOS OTA update job

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

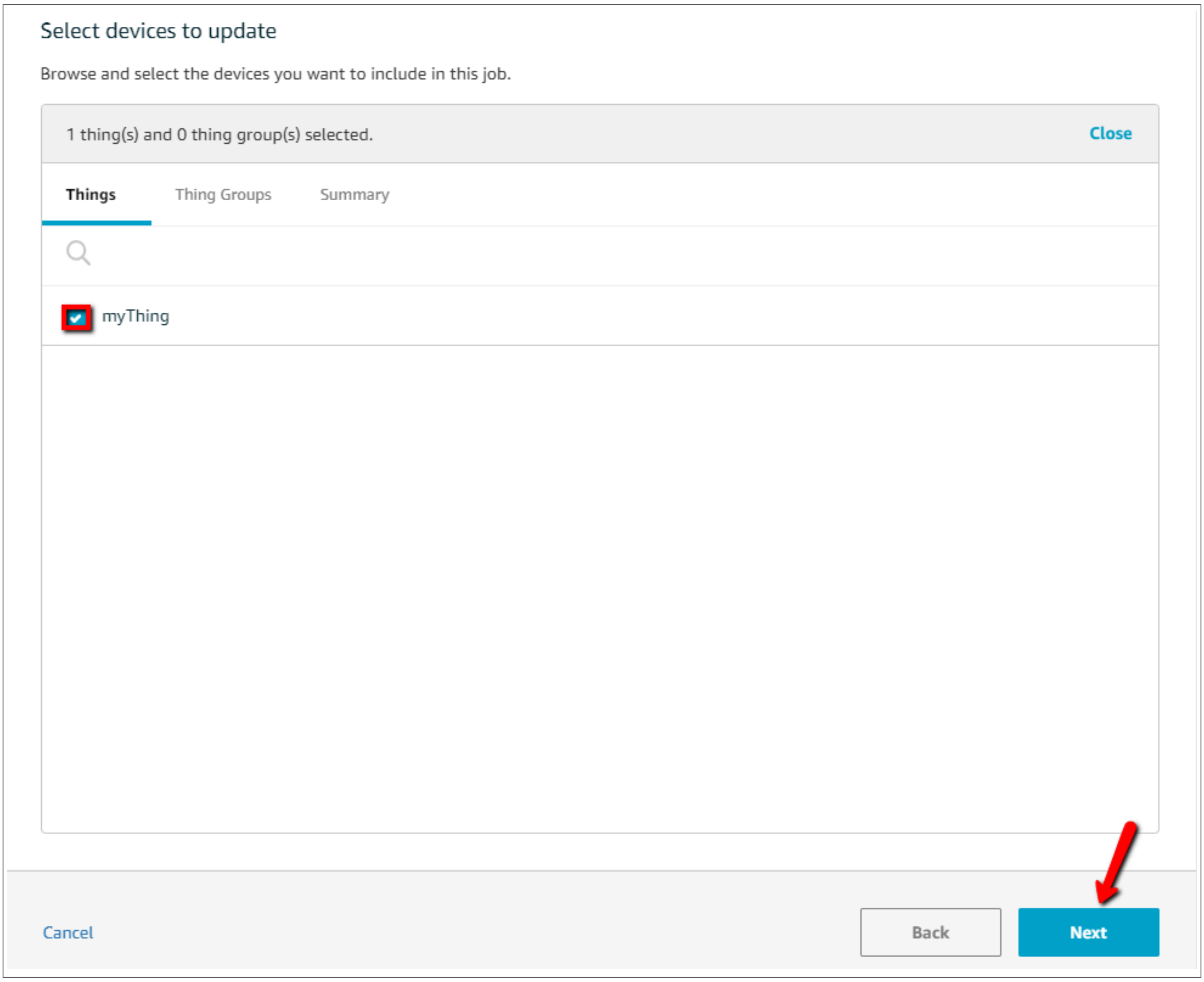
Select devices to update

Browse and select the devices you want to include in this job.

No devices or thing groups selected

□ myThing

6. Select the checkbox next to the IoT thing associated with your device. Choose **Next**.



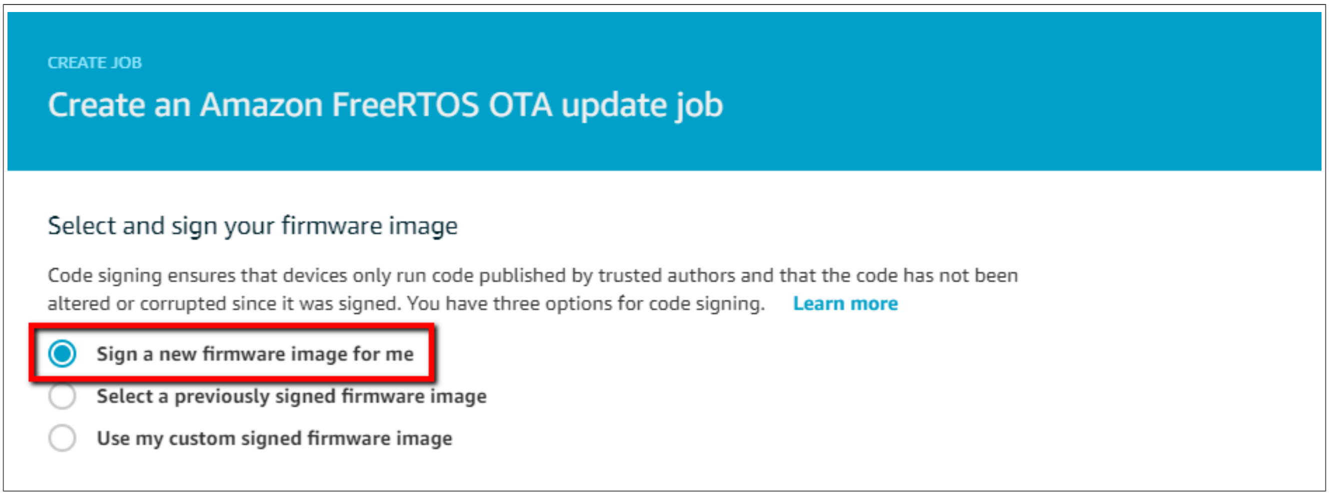
Select devices to update

Browse and select the devices you want to include in this job.

1 thing(s) and 0 thing group(s) selected.

☑ myThing

7. Under **Select and sign your firmware image**, choose **Sign a new firmware image for me**.



CREATE JOB

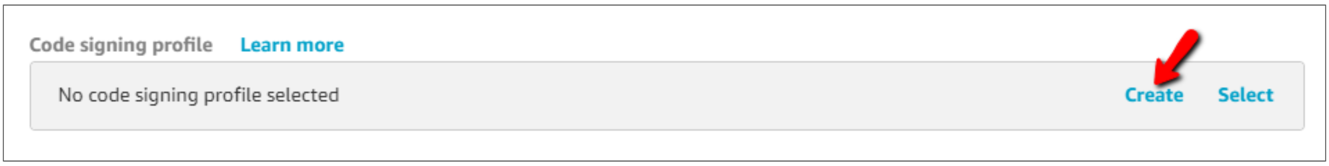
Create an Amazon FreeRTOS OTA update job

Select and sign your firmware image

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. Learn more

● Sign a new firmware image for me
○ Select a previously signed firmware image
○ Use my custom signed firmware image

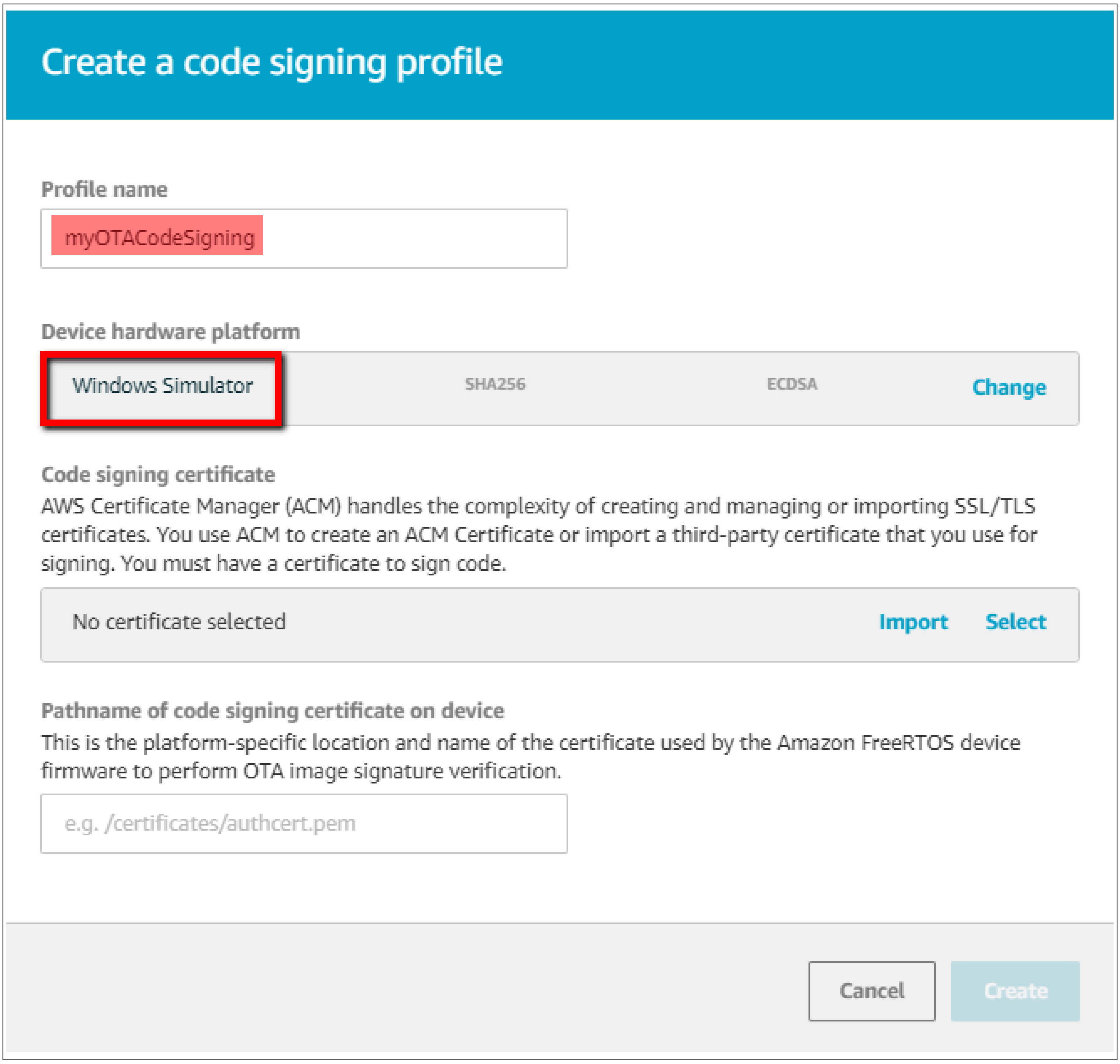
8. Under **Code signing profile**, choose **Create**.



Code signing profile Learn more

No code signing profile selected

9. In **Create a code signing profile**, enter a name for your code-signing profile.
   a. Under **Device hardware platform**, select Windows Simulator.



Create a code signing profile

Profile name

myOTACodeSigning

Device hardware platform

Windows Simulator SHA256 ECDSA

Code signing certificate

AWS Certificate Manager (ACM) handles the complexity of creating and managing or importing SSL/TLS certificates. You use ACM to create an ACM Certificate or import a third-party certificate that you use for signing. You must have a certificate to sign code.

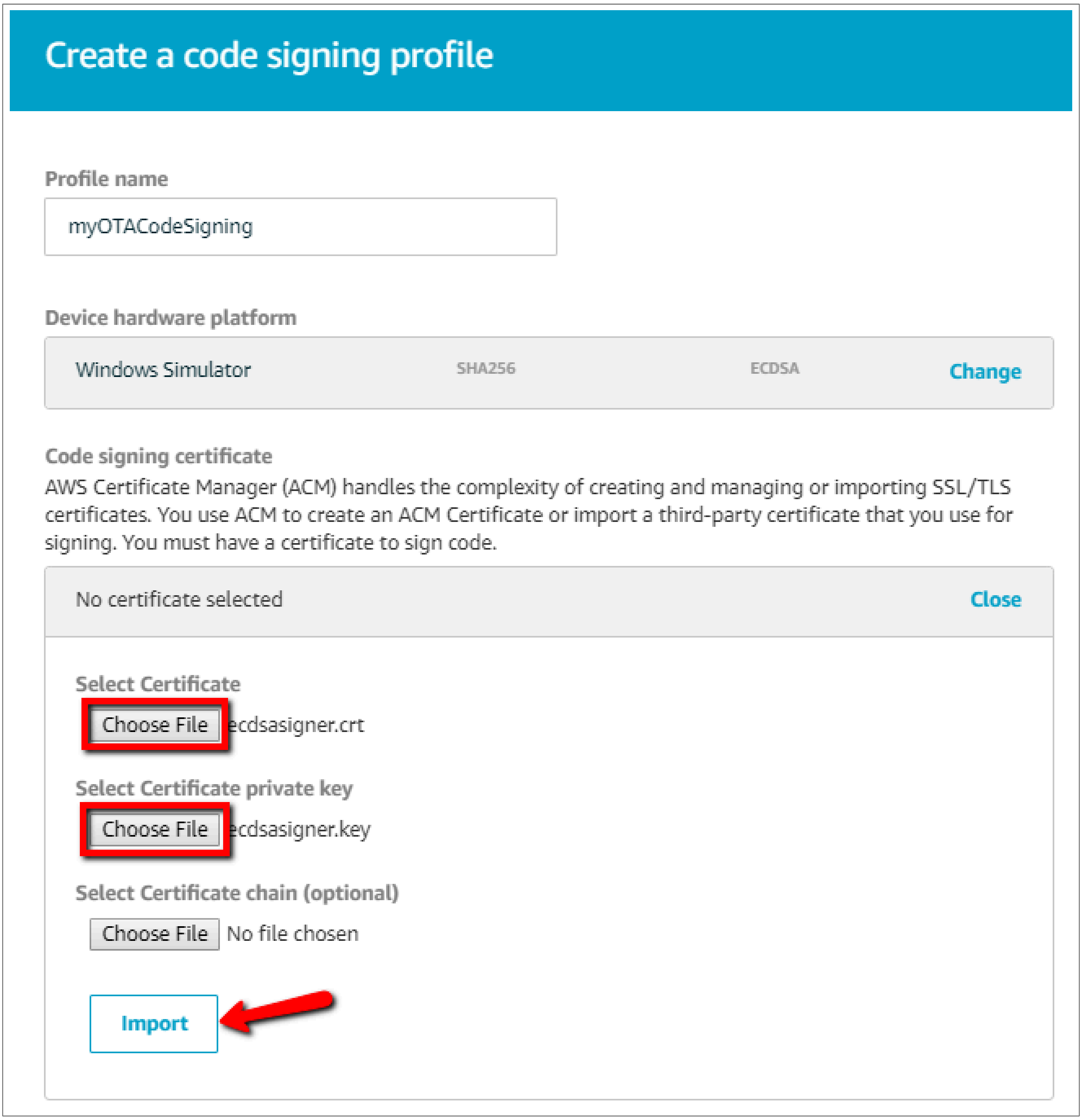
No certificate selected

Pathname of code signing certificate on device

This is the platform-specific location and name of the certificate used by the Amazon FreeRTOS device firmware to perform OTA image signature verification.

e.g. /certificates/authcert.pem

   b. Under **Code signing certificate**, choose **Import** and browse for the ecda certificate created with AWS CLI.



Create a code signing profile

Profile name

myOTACodeSigning

Device hardware platform

Windows Simulator SHA256 ECDSA

Code signing certificate

AWS Certificate Manager (ACM) handles the complexity of creating and managing or importing SSL/TLS certificates. You use ACM to create an ACM Certificate or import a third-party certificate that you use for signing. You must have a certificate to sign code.

No certificate selected

Select Certificate

ecdsasigner.crt

Select Certificate private key

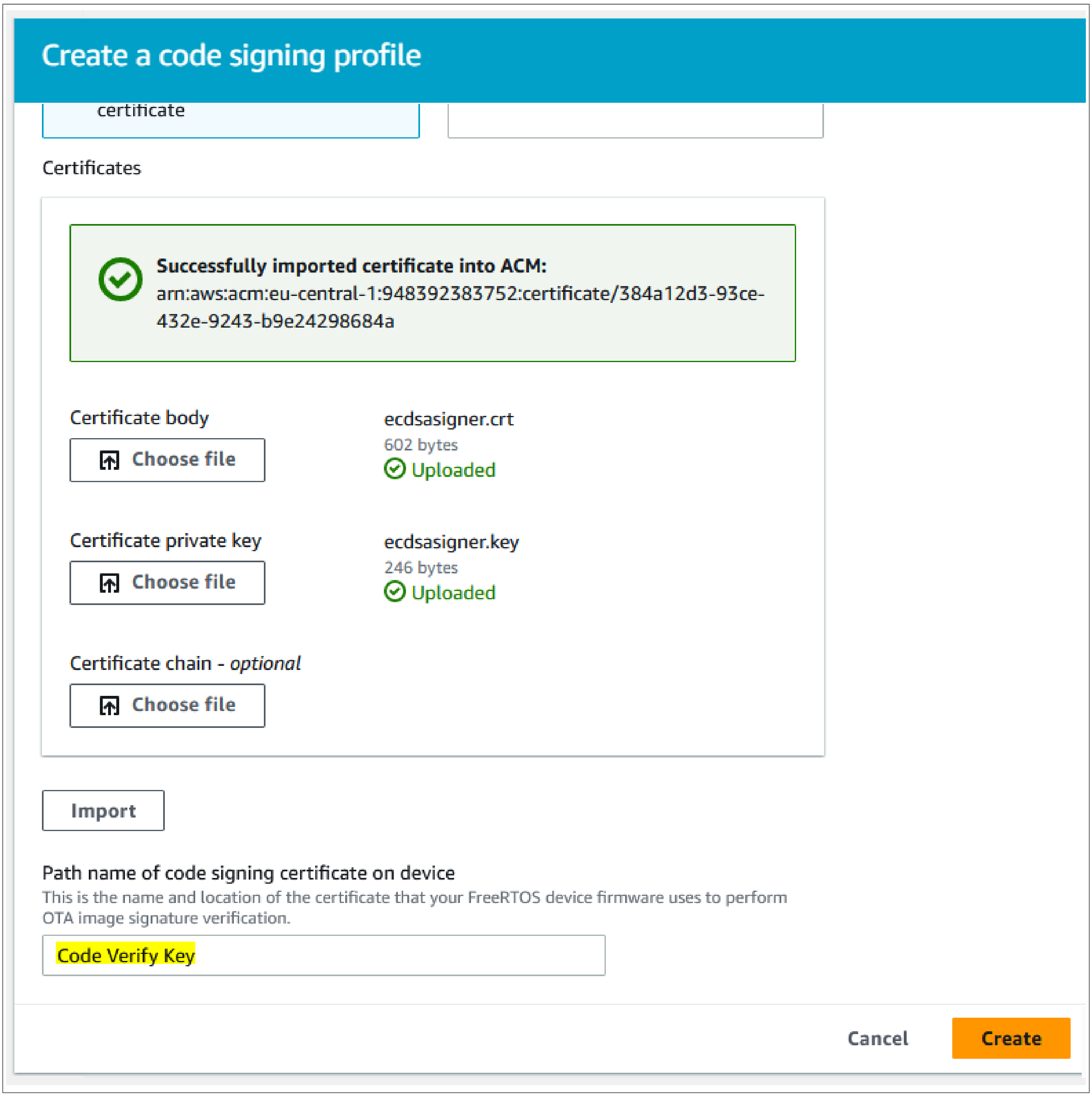
ecdsasigner.key

Select Certificate chain (optional)

No file chosen

c. Under **Pathname of code signing certificate on device**, type: Code Verify Key; must align with the pkcs11configLABEL_CODE_VERIFICATION_KEY defined in core_pkcs11_config.h.

d. click **Create**.



Create a code signing profile

certificate

Certificates

Successfully imported certificate into ACM:
arn:aws:acm:eu-central-1:948392383752:certificate/384a12d3-93ce-432e-9243-b9e24298684a

Certificate body — ecdsasigner.crt, 602 bytes, Uploaded

Certificate private key — ecdsasigner.key, 246 bytes, Uploaded

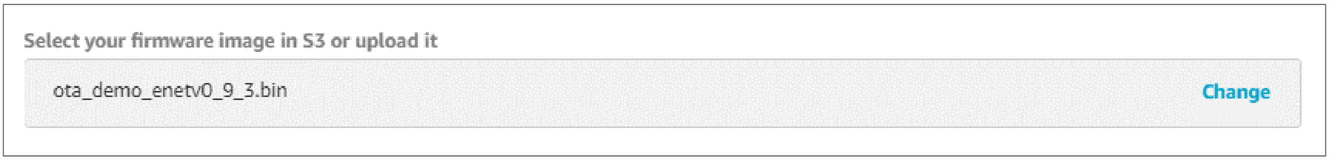
Certificate chain - *optional*

Path name of code signing certificate on device

This is the name and location of the certificate that your FreeRTOS device firmware uses to perform OTA image signature verification.

Code Verify Key

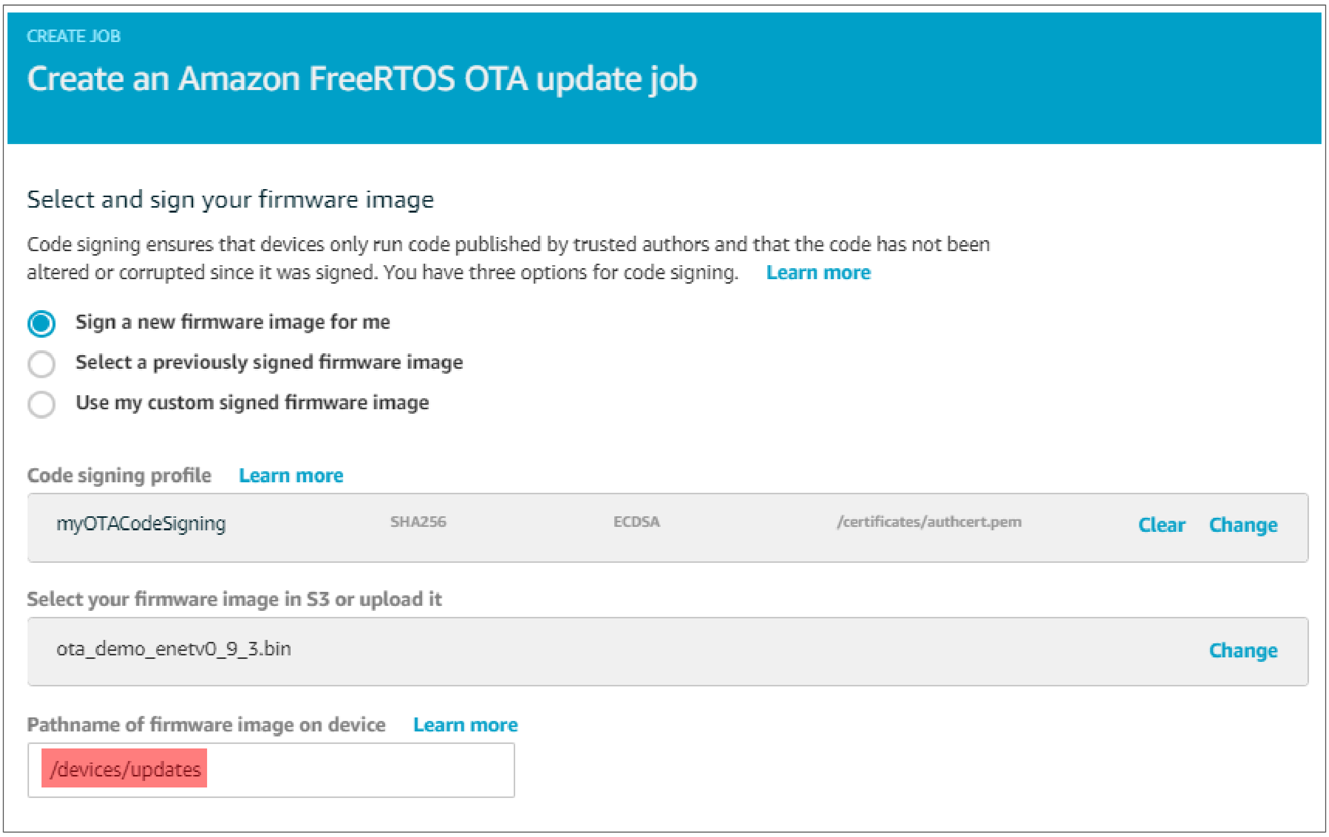
10. Under **Select your firmware image in S3**, choose **Select**.



Select your firmware image in S3 or upload it

ota_demo_enetv0_9_3.bin

11. Under **Pathname of firmware image on device**, type the default path /device/updates.



CREATE JOB

Create an Amazon FreeRTOS OTA update job

Select and sign your firmware image

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. Learn more

● Sign a new firmware image for me
○ Select a previously signed firmware image
○ Use my custom signed firmware image

Code signing profile Learn more

myOTACodeSigning SHA256 ECDSA /certificates/authcert.pem

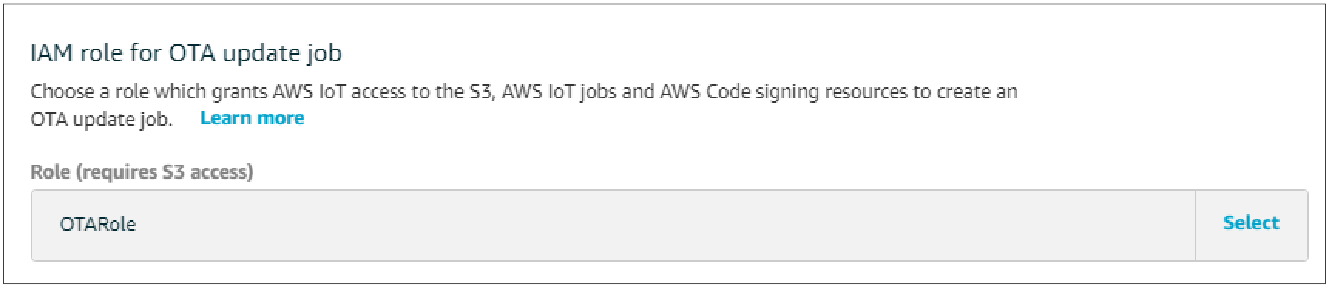
Select your firmware image in S3 or upload it

ota_demo_enetv0_9_3.bin

Pathname of firmware image on device Learn more

/devices/updates

12. Under **IAM role for OTA update job**, choose the role created in previous steps.



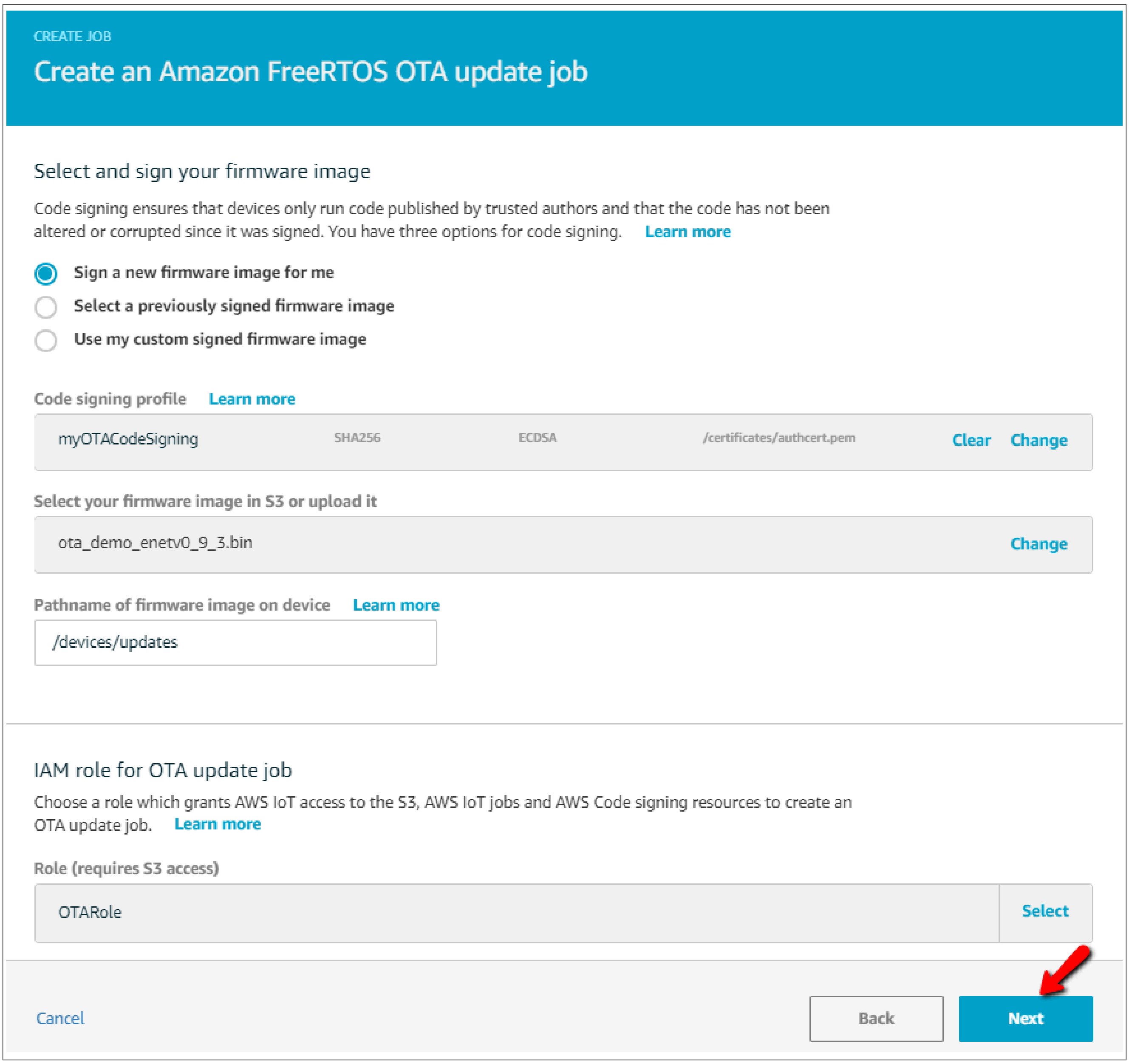
IAM role for OTA update job

Choose a role which grants AWS IoT access to the S3, AWS IoT jobs and AWS Code signing resources to create an OTA update job. Learn more

Role (requires S3 access)

OTARole

13. Choose **Next**.



CREATE JOB

Create an Amazon FreeRTOS OTA update job

Select and sign your firmware image

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. Learn more

● Sign a new firmware image for me

○ Select a previously signed firmware image

○ Use my custom signed firmware image

Code signing profile Learn more

myOTACodeSigning SHA256 ECDSA /certificates/authcert.pem Clear Change

Select your firmware image in S3 or upload it

ota_demo_enetv0_9_3.bin Change

Pathname of firmware image on device Learn more

/devices/updates

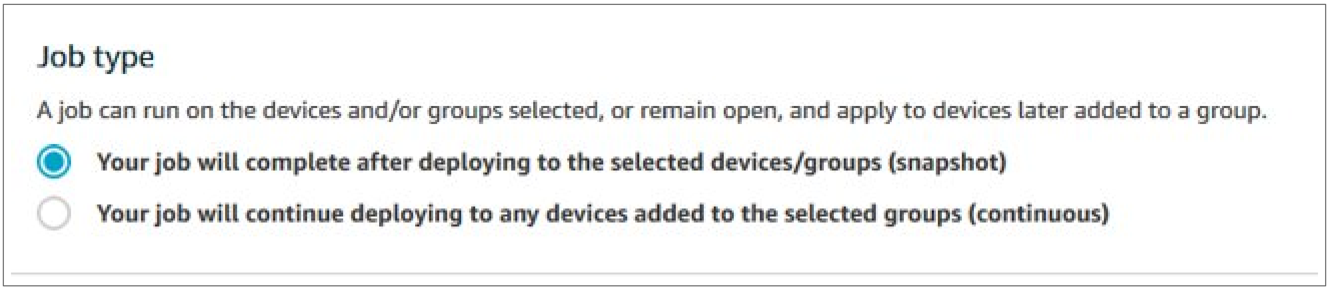
IAM role for OTA update job

Choose a role which grants AWS IoT access to the S3, AWS IoT jobs and AWS Code signing resources to create an OTA update job. Learn more

Role (requires S3 access)

OTARole Select

14. Under **Job type**, choose **Your job will complete after deploying to the selected devices/groups (snapshot)**.



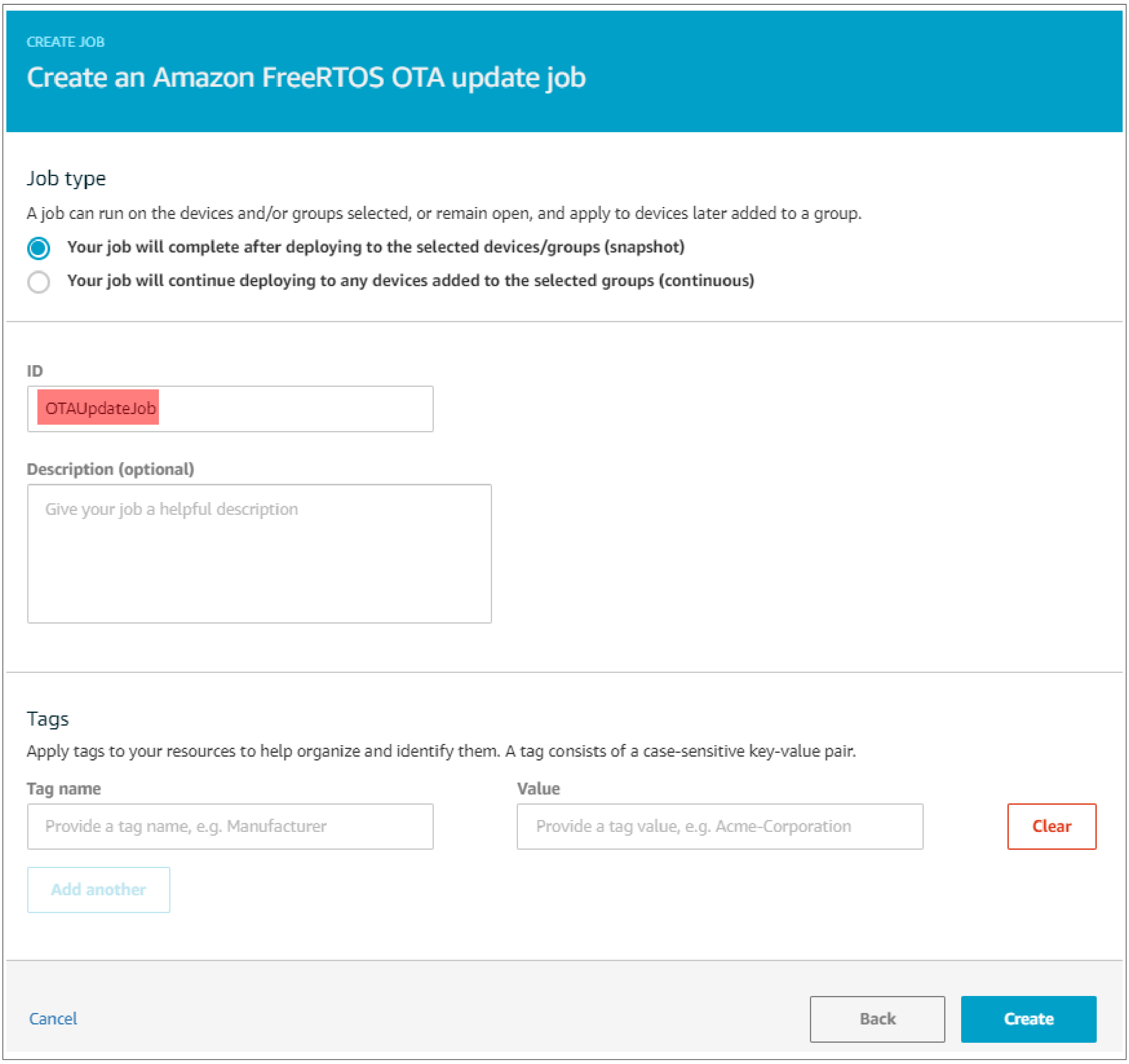
Job type

A job can run on the devices and/or groups selected, or remain open, and apply to devices later added to a group.

● Your job will complete after deploying to the selected devices/groups (snapshot)

○ Your job will continue deploying to any devices added to the selected groups (continuous)

15. Enter an ID for your OTA update job that the application must run, before clicking the **Create** button.



CREATE JOB

Create an Amazon FreeRTOS OTA update job

Job type

A job can run on the devices and/or groups selected, or remain open, and apply to devices later added to a group.

● **Your job will complete after deploying to the selected devices/groups (snapshot)**

○ **Your job will continue deploying to any devices added to the selected groups (continuous)**

ID: OTAUpdateJob

Description (optional): Give your job a helpful description

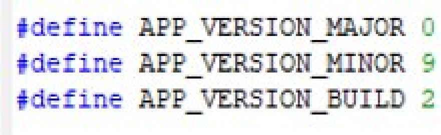
Tags

Apply tags to your resources to help organize and identify them. A tag consists of a case-sensitive key-value pair.

Tag name	Value	
Provide a tag name, e.g. Manufacturer	Provide a tag value, e.g. Acme-Corporation	Clear

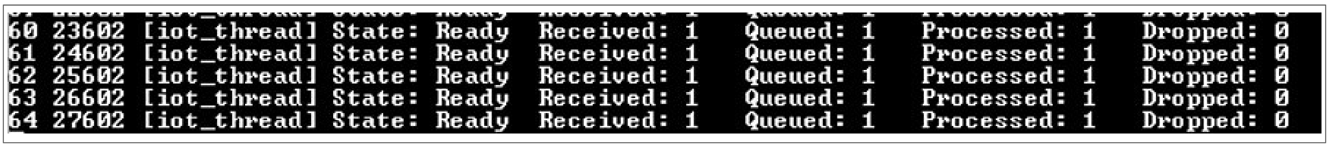
6.4 Running the application

1. Open `ota_config.h`.
2. Change any of the APP_VERSION macros to the original value.



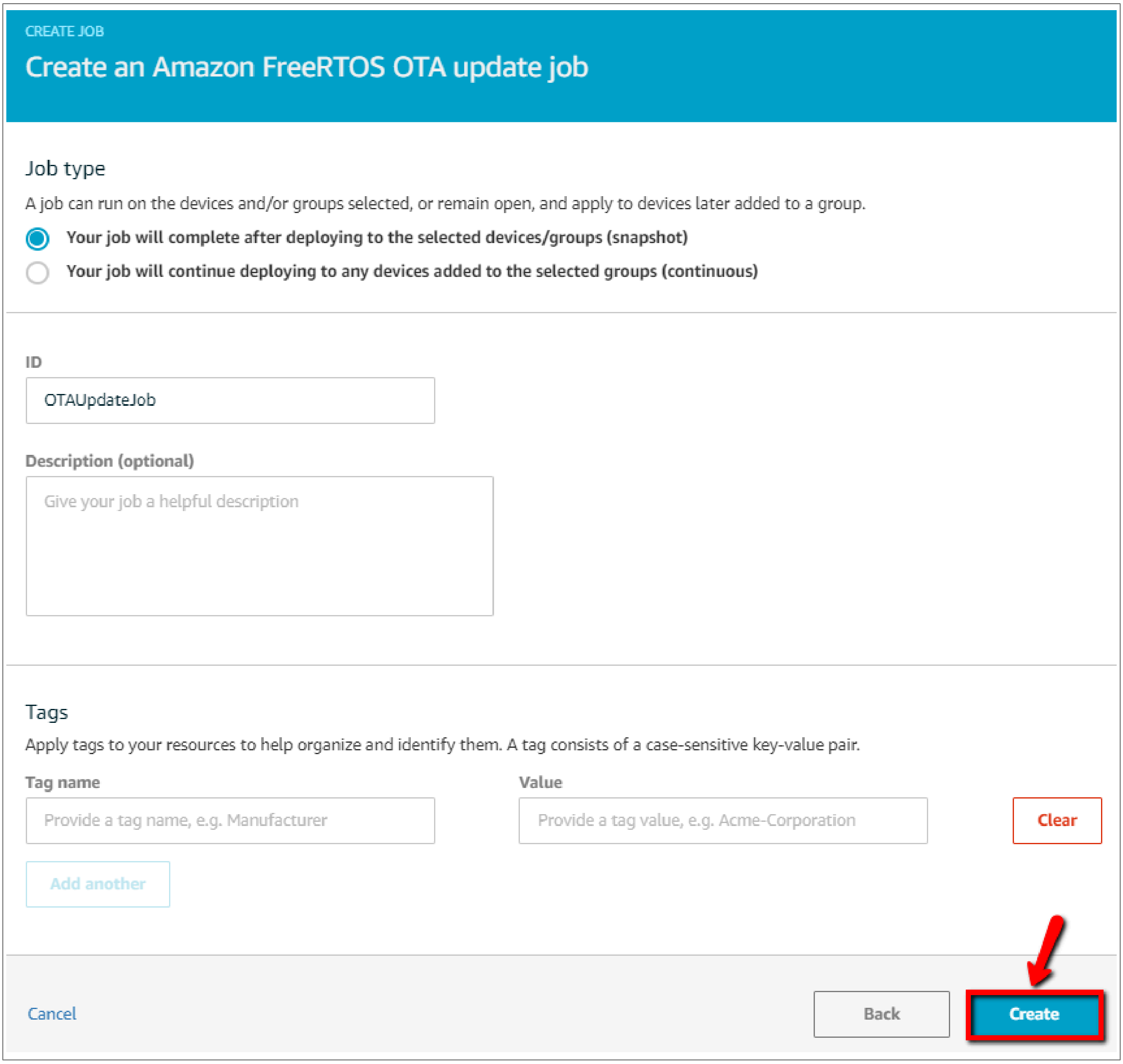
```
#define APP_VERSION_MAJOR 0
#define APP_VERSION_MINOR 9
#define APP_VERSION_BUILD 2
```

3. Make and Download and Debug.
4. When running the application, wait until the message of the OTA State Ready is shown in the serial terminal.



```
60 23602 [iot_thread] State: Ready  Received: 1   Queued: 1   Processed: 1   Dropped: 0
61 24602 [iot_thread] State: Ready  Received: 1   Queued: 1   Processed: 1   Dropped: 0
62 25602 [iot_thread] State: Ready  Received: 1   Queued: 1   Processed: 1   Dropped: 0
63 26602 [iot_thread] State: Ready  Received: 1   Queued: 1   Processed: 1   Dropped: 0
64 27602 [iot_thread] State: Ready  Received: 1   Queued: 1   Processed: 1   Dropped: 0
```

5. The OTA agent waits for an OTA job. Go back to the Create OTA job window and click **Create**.



CREATE JOB

Create an Amazon FreeRTOS OTA update job

Job type

A job can run on the devices and/or groups selected, or remain open, and apply to devices later added to a group.

● Your job will complete after deploying to the selected devices/groups (snapshot)

○ Your job will continue deploying to any devices added to the selected groups (continuous)

ID

OTAUpdateJob

Description (optional)

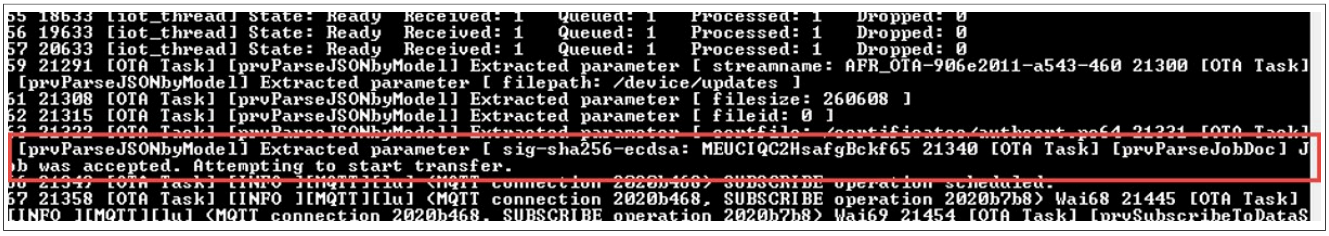
Give your job a helpful description

Tags

Apply tags to your resources to help organize and identify them. A tag consists of a case-sensitive key-value pair.

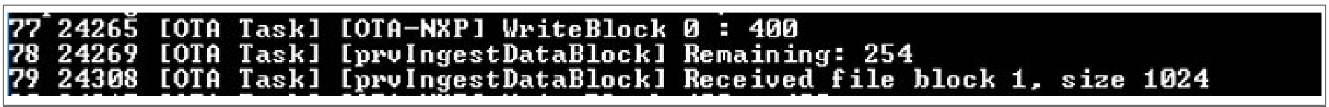
Tag name	Value	
Provide a tag name, e.g. Manufacturer	Provide a tag value, e.g. Acme-Corporation	Clear

6. The process starts, you can see a similar output.

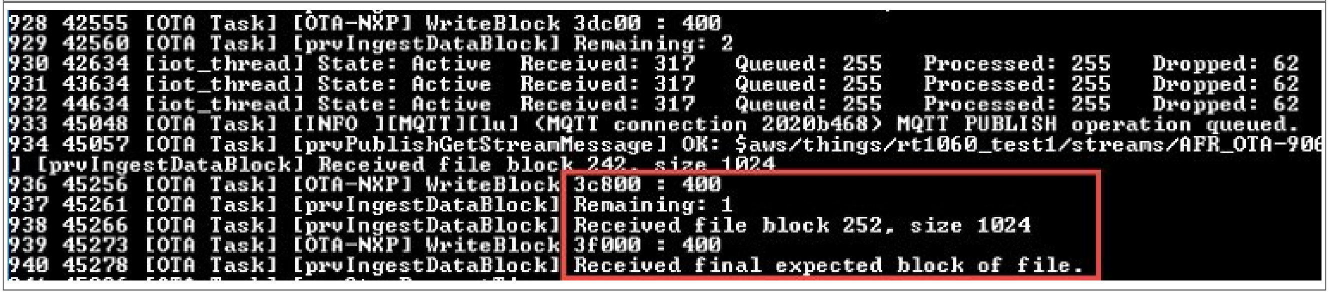


```
55 18633 [iot_thread] State: Ready  Received: 1   Queued: 1   Processed: 1   Dropped: 0
56 19633 [iot_thread] State: Ready  Received: 1   Queued: 1   Processed: 1   Dropped: 0
57 20633 [iot_thread] State: Ready  Received: 1   Queued: 1   Processed: 1   Dropped: 0
59 21291 [OTA Task] [prvParseJSONbyModel] Extracted parameter [ streamname: AFR_OTA-906e2011-a543-460 21300 [OTA Task]
 [prvParseJSONbyModel] Extracted parameter [ filepath: /device/updates ]
61 21308 [OTA Task] [prvParseJSONbyModel] Extracted parameter [ filesize: 260608 ]
62 21315 [OTA Task] [prvParseJSONbyModel] Extracted parameter [ fileid: 0 ]
63 21322 [OTA Task] [prvParseJSONbyModel] Extracted parameter [ certfile: /certificates/authcert.pe64 21331 [OTA Task]
 [prvParseJSONbyModel] Extracted parameter [ sig-sha256-ecdsa: MEUCIQC2HsafgBckf65 21340 [OTA Task] [prvParseJobDoc] J
ob was accepted. Attempting to start transfer.
66 21349 [OTA Task] [INFO ][MQTT][lu] <MQTT connection 2020b468> SUBSCRIBE operation scheduled.
67 21358 [OTA Task] [INFO ][MQTT][lu] <MQTT connection 2020b468, SUBSCRIBE operation 2020b7b8> Wai68 21445 [OTA Task]
[INFO ][MQTT][lu] <MQTT connection 2020b468, SUBSCRIBE operation 2020b7b8> Wai69 21454 [OTA Task] [prvSubscribeToDataS
```

7. Start file transfer.

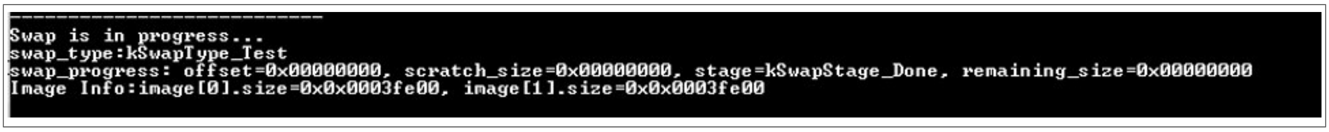


```
77 24265 [OTA Task] [OTA-NXP] WriteBlock 0 : 400
78 24269 [OTA Task] [prvIngestDataBlock] Remaining: 254
79 24308 [OTA Task] [prvIngestDataBlock] Received file block 1, size 1024
```



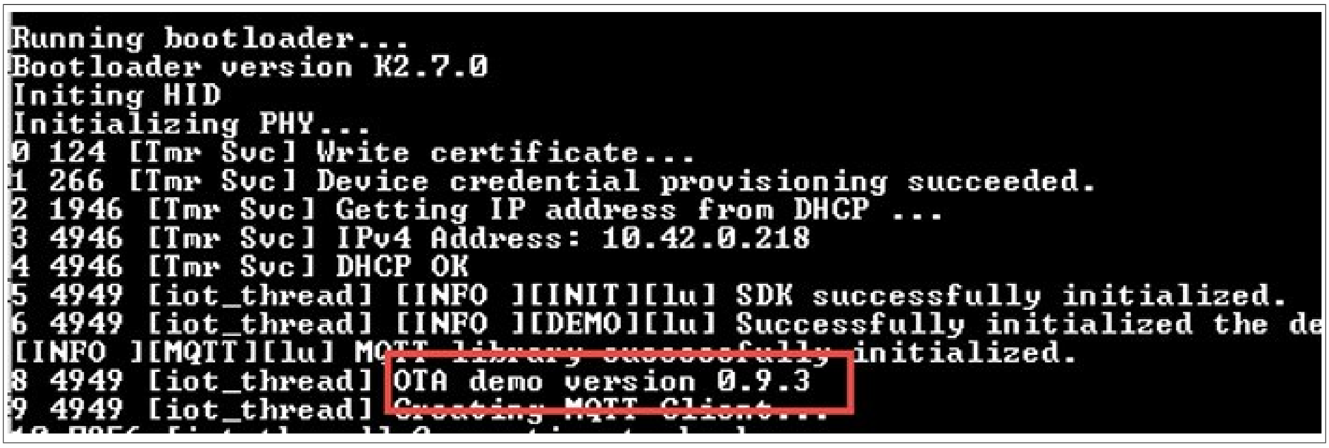
```
928 42555 [OTA Task] [OTA-NXP] WriteBlock 3dc00 : 400
929 42560 [OTA Task] [prvIngestDataBlock] Remaining: 2
930 42634 [iot_thread] State: Active  Received: 317  Queued: 255  Processed: 255  Dropped: 62
931 43634 [iot_thread] State: Active  Received: 317  Queued: 255  Processed: 255  Dropped: 62
932 44634 [iot_thread] State: Active  Received: 317  Queued: 255  Processed: 255  Dropped: 62
933 45048 [OTA Task] [INFO ][MQTT][lu] <MQTT connection 2020b468> MQTT PUBLISH operation queued.
934 45057 [OTA Task] [prvPublishGetStreamMessage] OK: $aws/things/rt1060_test1/streams/AFR_OTA-906
] [prvIngestDataBlock] Received file block 242, size 1024
936 45256 [OTA Task] [OTA-NXP] WriteBlock 3c800 : 400
937 45261 [OTA Task] [prvIngestDataBlock] Remaining: 1
938 45266 [OTA Task] [prvIngestDataBlock] Received file block 252, size 1024
939 45273 [OTA Task] [OTA-NXP] WriteBlock 3f000 : 400
940 45278 [OTA Task] [prvIngestDataBlock] Received final expected block of file.
```

8. Swap.



```
----------------------------
Swap is in progress...
swap_type:kSwapType_Test
swap_progress: offset=0x00000000, scratch_size=0x00000000, stage=kSwapStage_Done, remaining_size=0x00000000
Image Info:image[0].size=0x0x0003fe00, image[1].size=0x0x0003fe00
```

9. Device gets restarted, then the new application starts running.



```
Running bootloader...
Bootloader version K2.7.0
Initing HID
Initializing PHY...
0 124 [Tmr Svc] Write certificate...
1 266 [Tmr Svc] Device credential provisioning succeeded.
2 1946 [Tmr Svc] Getting IP address from DHCP ...
3 4946 [Tmr Svc] IPv4 Address: 10.42.0.218
4 4946 [Tmr Svc] DHCP OK
5 4949 [iot_thread] [INFO ][INIT][lu] SDK successfully initialized.
6 4949 [iot_thread] [INFO ][DEMO][lu] Successfully initialized the de
[INFO ][MQTT][lu] MQTT library successfully initialized.
8 4949 [iot_thread] OTA demo version 0.9.3
9 4949 [iot_thread] Creating MQTT Client...
```

7 Revision history

This table summarizes revisions to this document.

Table 1. Revision history

Revision number	Date	Substantive changes
0	12/2019	Initial release
1	01 June 2020	Updated for MCUXpresso SDK v2.8.0
2	15 June 2022	Layout updated for MCUXpresso SDK v2.12.0. Added a note in the overview section and made some editorial changes.
3	27 July 2023	Updated for MCUXpresso SDK v2.14.0 for changes in the ota_demo application.
4	10 January 2024	Updated for MCUXpresso SDK v2.15.000.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Amazon Web Services, AWS, the Powered by AWS logo, and FreeRTOS — are trademarks of Amazon.com, Inc. or its affiliates.

i.MX — is a trademark of NXP B.V.

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

For more information, please visit: https://www.nxp.com