

Onboarding-by-Claim Customer/OEM Guide

AWS IoT ExpressLink



Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS IoT ExpressLink: Onboarding-by-Claim Customer/OEM Guide

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

AWS IoT ExpressLink Onboarding-by-Claim Customer/OEM Guide	1
Overview	1
The onboarding process	1
Enable onboarding-by-claim	3
Create a registration application	3
Claim implementation	4
Configure the OEM account	. 4
Summary	4
Appendix A - Steps to obtain a claim-thing	5
Generate a "claim-thing" certificate	5
Register as an OEM with AWS	8
Appendix B - Register the ExpressLink manufacturer certificate authority (CA)	9
Appendix C - Create a JITP template 1	10
Appendix D - Glossary 1	19

AWS IoT ExpressLink Onboarding-by-Claim Customer/ OEM Guide

AWS IoT ExpressLink modules are hardware modules that enable easy cloud connectivity and security for AWS IoT devices. OEMs can integrate ExpressLink modules into their products to accelerate IoT development. ExpressLink modules come pre-provisioned with unique credentials to authenticate with AWS IoT Core. There are several options for onboarding ExpressLink devices, including individual/batch uploads, just-in-time provisioning, and fleet provisioning. The description focuses on a novel "onboarding-by-claim" process specific to ExpressLink. This leverages the module's unique capabilities to provide a more secure and streamlined onboarding experience.

Overview

AWS IoT <u>ExpressLink</u> modules are hardware connectivity modules that enable easy AWS cloud connectivity and implement strict and AWS-mandated security requirements for device-to-cloud connections. OEMs can accelerate the development of IoT products by integrating ExpressLink modules into their designs.

ExpressLink modules come pre-provisioned with a unique identifier and a certificate signed by the module manufacturer's Certificate Authority (CA), ready to authenticate with AWS IoT Core. *Onboarding* refers to the act of binding the module's credentials to a thing inside the <u>AWS IoT</u> registry of an OEM's account. There are multiple ways to onboard devices:

- individual certificate upload
- batch certificate upload
- just-in-time provisioning (JITP)
- just-in-time registration (JITR)
- fleet provisioning

The onboarding process

This guide describes a novel *onboarding-by-claim* mechanism specifically created to leverage an ExpressLink module's unique capabilities.

By default, ExpressLink modules connect to ExpressLink *staging account endpoints*. The staging account is managed by AWS to facilitate the onboarding-by-claim mechanism. It acts as a spring board to dispatch devices to their ultimate destination– the customer/OEM account. (See the <u>AWS</u> IoT ExpressLink Getting Started Guide).

The onboarding-by-claim process uses the <u>Just-in-time provisioning (JITP)</u> mechanism to automatically upload the device certificate, associate a policy, and create a "thing", but provides additional features, including:

- Late binding– the onboarding happens only when the end-user activates a finished product. This makes the onboarding process less time consuming during product manufacturing and, therefore, less expensive.
- No disclosure of confidential information is required with any element of the supply chain. This
 makes the process more secure and flexible, as the supply chain of trust is reduced to a direct
 link from the ExpressLink module manufacturer to the end-user in possession of the finished
 product.

The Onboarding Process in Detail: the user experience

- The onboarding-by-claim process is driven by the end-user who purchased a finished IoT capable product. It is triggered when the user interacts with a "product registration portal". This is OEM-specific software that can be a web or mobile application that offers end-users the opportunity to bind their identity to the unique product in their possession. After that, the OEM's application(s) can offer unique, personalized services augmented by the product's IoT connectivity.
- During product registration, the user is instructed to turn on the product and connect it for the first time. If the product uses an ExpressLink module, it connects to the default staging account, unless otherwise configured. Then, it automatically subscribes to a configuration (MQTT) topic.
- 3. The end-user is instructed to enter a unique identifier that they can find on a label on the finished product or its packaging. This identifier is a long alphanumerical string or possibly a QR code (the preferred option if the registration portal is implemented as a mobile app).
- 4. The identifier can now be used to find the unique "thing" present in the staging account and communicate with it (using the configuration topic) to provide the desired customer/OEM endpoint. It is this latter action that constitutes the actual "claim". After that, the device disconnects from the staging account, and then connects to the newly assigned endpoint.

5. The just-in-time provisioning (JITP) mechanism completes the process– the new device is authenticated, and a new "thing" is created in the selected customer/OEM account according to the instructions provided by a JITP template.

Enable onboarding-by-claim

In order for a product to take advantage of the onboarding-by-claim process, the OEM must ensure that the following components are available and properly configured:

- A registration (web) portal or mobile application.
- The claim-script this sends the new endpoint to the selected device inside the staging account.
- The customer/OEM account this must be properly configured to support just-in-time provisioning.

The following sections explain how these components operates, and describe how to properly configure, and provide implementation examples.

Create a registration application

The registration application is a software product that is (ideally) completely customized for the specific OEM product and brand. It requests the end-user to input the device's unique identifier, and then launches the claim-script that provides the device with a new target endpoint.

Note that the registration portal can be responsible for collecting additional end-user information, such as the end-user's location, and for including personally identifiable information that can be used by the application's location to select a target endpoint among several alterantives (if the OEM controls multiple AWS accounts). This also makes it possible to optimize the end-user experience and reduce latency (by selecting the most appopriate region, for example). The logic used, and any additional information optionally collected, are completely outside the scope of this document and are not essential to the onboarding process.

For an example implementation of a basic (web) registration portal, refer to the <u>claim provisioning</u> <u>reference implementation</u> (download).

Claim implementation

The actual claim is implemented in the claim-script (a function of the registration application). To do this, it publishes a specific, JSON formatted message on the unique device configuration topic. This operation requires the script to obtain access to the ExpressLink module staging account which is managed by AWS. While AWS does not share such credentials with customers/OEMs, upon request the AWS IoT Device service team allows the customer/OEM to create a "claim-thing" within the staging account registry. The claim-thing can then be controlled by the OEM registration application (using an MQTT client API) to publish the endpoint update message. To request and obtain control of a claim-thing follow the configuration steps indicated in <u>Appendix A - Steps to</u> <u>obtain a claim-thing</u>.

Configure the OEM account

In the last step, the customer/OEM's AWS account must be configured to enable the use of the just-in-time provisioning mechanism. To do this:

- 1. Register the ExpressLink module vendor's Certificate Authority with the customer/OEM account. Follow the steps in <u>Appendix B Register the ExpressLink manufacturer certificate</u> <u>authority (CA)</u>.
- Create a JITP template so that new devices that are directed to the account will be automatically associated with a desired policy and given a proper thing-name. Follow the steps in Appendix C - Create a JITP template.

Summary

To summarize, ExpressLink modules come pre-provisioned with a unique identifier and a certificate signed by the module manufacturer Certificate Authority (CA), ready to authenticate with AWS IoT Core.

Onboarding, the act of binding the module credentials to a <u>thing</u> inside the AWS IoT registry of an customer/OEM's account is accomplished using various mechanisms provided to all devices that connect to AWS IoT Core. This guide describes a novel onboarding-by-claim mechanism specifically created to leverage an ExpressLink module's unique capabilities.

By following the steps in this document, any customer/OEM can take advantage of this new capability to provide their own customers with the best experience, while optimizing the supply chain for security and flexibility.

Appendix A - Steps to obtain a claim-thing

This appendix talks about the steps to obtain a claim-thing. To generate a claim-thing certificate, follow this 7 step process. Once the certificate is generated, the next thing to do is to register as an OEM with AWS. This is a 3 step process which details information about the registration.

Generate a "claim-thing" certificate

- 1. If you do not have AWS account, or wish to use a new one specifically for ExpressLink follow the steps to <u>Create an AWS account</u>.
- 2. If you aren't already signed in to your AWS account, sign in, then open the <u>AWS IoT console</u>.

aws	Services	Q IoT Core	Х D Ф 0re	gon 🔻
			Search results for 'IoT Core'	٥
		Services (38)	Services	See all 38 results ►
		Features (78)		
		Resources New	Gennect Devices to the Cloud	
		Blogs (10,121)		
		Documentation (22,123)	AWS IoT Core for LoRaWAN	
		Knowledge Articles (30)	Connect, manage, and secure LoRaWAN devices at scale	
		Tutorials (34)		
		Events (183)		
		Marketplace (53)	Collect, preprocess, store, analyze and visualize data of 101 devices	
			≽ loT Events ☆	
			Monitor device fleets for changes and trigger alerts to respond	
			Features	See all 78 results ►
			Davisa Advisar	
Feedback	Looking for la	anguage selection? Find it in the new Unified	Settings 🔀 © 2022, Amazon Web Services, Inc.	or its affiliates. Privacy Terms Cookie preferences

3. In the AWS IoT console, on the left navigation pane, select **Security** to expand the sub-menu, then select **Certificates**.



4. On the **Certificates** page, on the right side of the table that shows currently-installed certificates, select **Add certificate**, then select **Create certificate** in the drop-down menu.

aws Services Q Search fo	r services, features, blogs, docs, and more	[Option+S]		Þ	¢	IsenLink	0	Oregon 🔻			
Connect many devices											٤
	AWS IOT > Security > Certificates										
Test	Certificates Info										
Device Advisor	X.509 certificates authenticate device and	client connections. Certificates must be	e registered with AWS IoT and a	ctivated before	a device	or client car	ı commu	nicate with AW	/S IoT.		
MQTT test client											
	Certificates Certificates you've t	transferred									
Manage									_		
All devices	Certificates (2)						C	Actions	Add	certificate 🔺	
Greengrass devices	O Find antifactor								Creat	te certificate	1
LPWAN devices									Regis	ster certificates	
Remote actions	Certificate ID		▼ Status	⊽ Crea	ited						~
Message Routing				_							
Retained messages											
Security											
Intro											
Certificates											
Policies											
Pole Aliases											
Authorizers											
Audit											
Detect											
Fleet Hub											
Device Software											
Feedback Looking for language selection	2 Find it in the new Unified Settings				@ 201	22 Amazon W	/eb Service	s Inc. or its affili	ates Privaci	v Terms Cor	okie preferences

5. On the **Create certificate** page, choose **Auto-generate new certificate**, and choose **Inactive**. Select **Create** to create an X.509 certificate.

Connect many devices	AWS INT > Servity > Certificate: > Create certificate				_	_
Test Device Advisor MQTT test client	Create certificate Info Certificates authenticate devices and clients so that they can connect to AWS IoT. Your device won't be able to connect to AWS IoT without authentication and an appropriate policy.					
Manage	Certificate					
All devices Greengrass devices LPWAN devices Remote actions Message Routing	Auto-generate new certificate (recommended) Generate a new certificate, public key, and private key using AWS IoT's certificate authority and register it with AWS IoT. Create certificate with certificate signing request (CSR) file to create and register a certificate tables do a private key you own.					
Retained messages						
 Security Intro Certificates 	Certificate status Assign the initial state of the new certificate. The certificate must be active before it can be used to connect to AWS IoT. You can change its status later in the certificate's detail page.					
Policies Certificate authorities Role Aliases Authorizers	Inactive A device wyn't be able to connect to AWS using this certificate until it's activated. A device will be able to connect to AWS using this certificate immediately after you create it.					
Audit Detect Fleet Hub 	Cancel)				
Device Software						

- 6. In the pop-up window that opens, select **Download** for each of the credentials files that you will need:
 - certificate fingerprint.pem.crt
 - certificate fingerprint-public.pem.key
 - certificate fingerprint-private.pem.key
 - Amazon Root CA 1 (this file will be downloaded as AmazonRootCA1.pem).

(The *certificate fingerprint* is a hexadecimal string that uniquely identifies the certificate and is generated using the certificate body.)



7. Select **Continue** to close the pop-up window, then store the keys and the certificate in a safe place following security best practices.

Register as an OEM with AWS

- Send an email with the following information to <expresslink-onboarding@amazon.com>:
 - Company name
 - AWS account ID
 - Technical/Developer Contact (name and email)
 - Technical Manager Contact (name and email)
- 2. When it receives the request, the AWS IoT ExpressLink service team will:
 - provide a secure mechanism for you to upload the certificate generated in the previous section.
 - create a <u>universally unique identifier</u> (UUID), a 128-bit string label for your onboarding functionality. The UUID is required to connect to the Staging Endpoint.

The AWS IoT ExpressLink service team will send the UUID for the onboarding functionality, instructions for uploading the certificate, related documentation, and terms & conditions to the two technical contacts listed in your request.

3. After you receive the information listed in the previous step, follow the instructions and upload the certificate (*certificate fingerprint*.pem.crt) that you generated in the previous section.

🔥 Warning

DO NOT upload the private key! (certificate fingerprint-private.pem.key).

Appendix B - Register the ExpressLink manufacturer certificate authority (CA)

After a claim-thing certificate is obtained, the next steps is to register the ExpressLink certificate with the manufacturer certificate authority (CA). This four step process walks the user through registering with the certificate authority (CA).

- 1. Follow the steps in <u>Getting started with the AWS CLI</u> to install the AWS CLI on your development machine.
- 2. Follow the steps in <u>Configuration and credential file settings</u> to configure the AWS CLI to use your AWS account credentials.
- 3. Register the root CA on your AWS account with the following AWS CLI command (replace *path-to-manufacturer-CA* with the local path of the root CA):

```
aws iot register-ca-certificate --ca-certificate file://path-to-manufacturer-CA --
certificate-mode SNI_ONLY --set-as-active --allow-auto-registration
```

(For more information on non-Amazon-signed certificates and certificate authorities on AWS IoT Core, see Create your own client certificates.

4. Record the *CA certificate id* that is shown in the output of the command above. The CA certificate id is a long hexadecimal string. You will need this later.

Appendix C - Create a JITP template

Create a JITP template so that new devices directed to the account are automatically associated with a desired policy, and given a proper thing-name on the AWS console. Follow the steps below to create a template.

1. Open the AWS IoT console.

	X D Q Oreg	gon 🔻
Services Q IoT Core Services (38) Features (78) Resources New Blogs (10,121) Documentation (22,123) Knowledge Articles (30) Tutorials (34)	× Image: A state of the constraint of	gon ▼ See all 38 results ▶
Events (183) Marketplace (53)	 Ø IoT Analytics ☆ Collect, preprocess, store, analyze and visualize data of IoT devices PoT Events ☆ Monitor device fleets for changes and trigger alerts to respond Features	See all 78 results ►
	Device Advisor	

2. In the left navigation pane, select **Connect many devices** then select **Provisioning templates** in the drop-down sub-menu.



3. On the provisioning templates management page, select **Create provisioning template**.

aws Services Q Search for se	for services, features, blogs, docs, and more [Option+S]	▷ 👃 IsenLink ⑦ Oregon 🕶
AWS IoT $\qquad \times$	X AWS IoT > Connect > Connect many devices	
Monitor	▼ How it works	
Connect Connect one device © Connect many devices Provisioning templates Bulk Registration	O O O O 1 1 1 0 0 0 0 0 0 O O O 0 0 0 1 1 1 1	
Test ▶ Device Advisor MQTT test client	Step 1. Determine provisioning scenario Step 2. Define device management structure Devices need a unique certificate to connect to AWS IoT. You can install this certificate during the device's manufacture, when the device is provisioned by an authenticated user, or by installing a claim certificate that's exchanged for a unique device certificate the first time the device connects to AWS IoT. Connected devices arr encorporative in AWS IoT by thing resources, whethey our organize, manage, and maintain your devices. Thing groups st thing groups st thing groups st thing you by thing the second and also be created when the device to AWS IoT. Learn more [2] Learn more [2]	Step 3. Create a provisioning template which A provisioning template is a JSON document that describes the resources, policies, and permissions to create for the device when it's provisioned. abo Learn more [2]
Manage All devices		
 Greengrass devices LPWAN devices 	Connect many devices (0) Info To connect many devices, the provisioning template automates the provisioning required to connect new devices.	Activate Deactivate Delete Create provisioning template
 Remote actions Message Routing 	Q. Find provisioning templates	< 1 > ©
Retained messages	Name ▲ Template type ▼ Created date ▼ State	tus 🗸
▼ Security Intro Certificates Policies Certificate authorities	No provisioning templates You don't have any provisioning templates in us-west-2. Create provisioning template	
Role Aliases Authorizers		

4. On the **Create provisioning template** page, choose **Provisioning devices with unique certificate (JITP) -** *recommended*, then select **Next**.

<page-header> Aws of model and the set of the s</page-header>	aws Services Q Search for se	rvices, features, blogs, docs, and more [Option+S]		▷ 수 IsenLink ⑦ Oregon ▼
<section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header>	AWS IOT ×	AWS IoT > Connect > Connect many devices > Create provisioning templat	e	
<section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><text><text><text><text><text><text><text><text><text><text><text><text><text></text></text></text></text></text></text></text></text></text></text></text></text></text></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header>	Monitor	Create provisioning template AWS IoT supports three device-provisioning scenarios to accommodate different d	evice manufacturing and installation processes. If you're not sure which scenario	to choose, refer to the developer guide for information.
<section-header><section-header><text><text><text><text><text><text><text><text><text><text><text><text></text></text></text></text></text></text></text></text></text></text></text></text></section-header></section-header>	Connect Connect one device Connect many devices	Provisioning scenario Choose the provisioning scenario that fits your device manufacturing and installation process	ses the best. Learn more 🕜	
 b Device Advisor MCT test dient Mange Al devices B Composition devices with unique device certificates Learn mer (2) C Torovision devices with unique device certificates Learn mer (2) C Mange B Composition devices with unique device certificates Learn mer (2) C Mange C Mange C Mange C Assage Routing <li< td=""><td>Provisioning templates Bulk Registration</td><td> Provisioning devices with unique certificates (JITP) - recommended Your IoT devices will be installed with unique device certificates already on the device. This scenario is also known as just-in-time provisioning (JITP). </td><td>Provisioning devices by authorized users Your to't devices don't have unique certificates when they are installed. Authorized installes or end users use an apt to provision the devices before they are consecuted to AWS lot. It his scenario, you provide the installation app to configure the device during installation and the device's firmware must support this providening process. This is also known as filter providening with</td><td>Provisioning devices with claim certificates Choose this option if your lo1 devices are delivered with claim certificates that are shared with other devices. The devices use their claim certificates to connect to WSIST for the first time. The claim certificate is replaced with a unique device certificate after provisioning. This option is also known as fleet provisioning with certificate.</td></li<>	Provisioning templates Bulk Registration	 Provisioning devices with unique certificates (JITP) - recommended Your IoT devices will be installed with unique device certificates already on the device. This scenario is also known as just-in-time provisioning (JITP). 	Provisioning devices by authorized users Your to't devices don't have unique certificates when they are installed. Authorized installes or end users use an apt to provision the devices before they are consecuted to AWS lot. It his scenario, you provide the installation app to configure the device during installation and the device's firmware must support this providening process. This is also known as filter providening with	Provisioning devices with claim certificates Choose this option if your lo1 devices are delivered with claim certificates that are shared with other devices. The devices use their claim certificates to connect to WSIST for the first time. The claim certificate is replaced with a unique device certificate after provisioning. This option is also known as fleet provisioning with certificate.
Nanage A II devices A Realand Ressauges V Security Intro Cartificate nust be one that you own, it must be registered with AWS IoT Non the CA certificate that signed the device certi	 Device Advisor MQTT test client 	To provision devices with unique device contificates luma	user.	
 Message Routing Retaind messages Security Security Security Certificates Policies Certificate authornies Rolass Authorizers 	Manage All devices Greengrass devices LPWAN devices Remote actions			
 Security certificate must be one that you own, it must be registered with AWS IoT in your account and Region, and it must have automatic certificate must be registered with AWS IoT in your account and Region, and it must have automatic certificate must be registered with AWS IoT in your account and Region, and it must have automatic certificate must be registered with AWS IoT in your account and Region, and it must have automatic certificate must be registered with AWS IoT in your account and Region, and it must have automatic certificate must be registered with AWS IoT must be registered with AWS IoT in your account and Region, and it must have automatic certificate must be registered with AWS will create for your device in a provisioning template that it connects to AWS IoT concurses and permissions that AWS will create for your device in a provision your device in a provisioned, it connects to AWS IoT concurses and communicates with AWS IoT in your account and Region, and it must have automatic certificate must be registered with AWS will create for your device in a provisioning template that it connects to AWS IoT concurses and permissions that AWS will create for your device when it connects. Certificate authorities Role Allases Authorizers 	 Message Routing Retained messages 	1. Choose the CA certificate Choose the CA certificate that signed the device certificates. The CA	2. Set provisioning actions Configure how AWS IoT should provision your IoT device when it first	3. Connect devices When your IoT devices first connect to AWS IoT, they'll be provisioned in
Policies Certificate authorities Role Allases Authorizers	 Security Intro Certificates 	certificate must be one that you own, it must be registered with AWS IoT in your account and Region, and it must have automatic certificate registration turned on.	connects to AWS IoT. You describe the AWS IoT resources and permissions that AWS will create for your device in a provisioning template that provisions your device when it connects.	AWS IoT according the provisioning template that you created. After the device is provisioned, it connects and communicates with AWS IoT normally.
Role Aliases Authorizers	Policies Certificate authorities			Cancel
	Role Aliases Authorizers			

- 5. In the JITP template creation wizard, under **Describe provisioning template**, enter the information for the **Provisioning template properties**:
 - a. Under **Provisioning template status**, choose **Active**.
 - b. Enter a **Provisioning template name**.
 - c. (Optional) Enter a **Description** for the template.

Step 1 Describe provisioning	Describe provisioning template Info
template	The details on this page describe the general aspects of the provisioning template that you're creating.
Step 2 Set provisioning actions	Provisioning template properties Info
Step 3	Provisioning template status The provisioning template status determines whether the template can be used to provision a new device. Only active templates can
Set device permissions	provision devices.
Step 4	 Inactive Inactive templates can't provision any devices that are configured to use it. You can create an
Review and create	 inactive template to prevent devices from being provisioned until you're ready. Active An active template can provision the devices that are configured to use it.
	Provisioning template name
	<your_template_name></your_template_name>
	The name can have up to 36 characters and must not contain spaces. Valid characters: A-Z, a-z, 0-9, and _ (underscore) and - (hyphen).
	Description - optional
	<pre><your_template_description></your_template_description></pre>
	472 characters compining

6. Under **Provisioning role**, make sure **Attach managed policy to IAM role** is checked. (This ensures the IAM role created here and used in device provisioning will have the needed privileges.) Then select **Create new role**.

Provisioning role			
The provisioning role uses an IAM role that	t authorizes AWS IoT to acces	s resources on your behal	lf.
Choose an IAM role	• (J View Z	Create new role
Attach managed policy to IAM ro	ole		

(Optional) Instead of creating a new role, you can choose a role that you have previously made. However, to make sure that the role has enough privileges to provision your ExpressLink modules, you must make sure that the role has the AWS managed policies "AWSIoTThingsRegistration", "AWSIoTLogging", and "AWSIoTRuleActions" attached, or that it has an inline policy with equivalent or greater permissions. See <u>AWS managed</u> policies for AWS IoT for more information.

7. In the **Create role** pop-up window, enter a **Role name**, then select **Create**.

Create role	×
The provisioning role uses an IAM role that authorizes AWS IoT to access resources on your behalf.	
Role name	
Cancel	

8. Under **CA certificate configuration**, for **Automatic certificate registration**, choose **On**. Above that, under **CA certificate** select the **Choose the CA certificates to use** dropdown menu.

CA certificate configuration Info To provision devices that have their unique device certificates before the device is installed, provisioning templates are associated with the CA certificate that signed the device certificate.
CA certificate The template to attach to the selected CA certificates.
Choose the CA certificates to use. View Register new CA Image: Cartificate registration
When turned on, certificates signed by this CA will be registered automatically. This must be turned on for provisioning templates to automatically provision devices with certificates signed by this CA.
 Off Device certificates signed by these CAs won't be registered automatically when they're first used to connect to AWS IoT.
On Device certificates signed by these CAs will be registered automatically when they're first used to connect to AWS IoT.

9. In the dropdown menu, choose the checkbox in front of the CA certificate ID that was listed in the output of the aws iot register-ca-certificate AWS CLI command that you ran in the previous section. After the CA certificate ID appears on the page, select **Next**.

CA certificate

The template to attach to the selected CA certificates.

id of the CA that you registered previously
ficates before the device is installed, provisioning templates are associated with the
ou registered previously
registered automatically. This must be turned on for provisioning templates to I by this CA.
gistered automatically when they're first used to
stered automatically when they're first used to

10. Under **Set provisioning actions**, toggle on **Automatically create a thing resource when provisioning a device**.

(Optional) You can also choose **Additional configurations** for the <u>Thing type</u>, <u>Searchable</u> thing attributes, <u>Thing groups</u>, and <u>Billing groups</u>.

Select Next.



11. Under **Set device permissions**, select **Create policy** to create a new policy for those ExpressLink modules you want to provision. This policy determines the actions that can be run by the ExpressLink modules on AWS IoT Core under your AWS account.

AWS IoT > Connect > Connect m Step 1 Describe provisioning template	Any devices > Create provisioning template > Provisioning by unique device certificates Set device permissions Info AWS IOT policies authorize devices to access AWS IoT resources such as other thing resources, MOTT topics, and Device		
Step 2 Set provisioning actions	Shadows.		
Step 3 Set device permissions	Policies (2) Info C Create policy [2] Choose up to 10 policies to attach to this certificate. < 1 > ③		
Step 4 Review and create	Policy name ▼ ARN ▼		
	Cancel Previous Next		

12. On the Create policy page, enter a Policy name.

On the **Policy statements** tab, under **Policy document**, for convenience you can enter "*" for both the **Policy action** and **Policy resource**. However, this policy will allow any and all actions on any AWS IoT Core resources accessible through MQTT. We recommend that you use a more restrictive policy. The **Policy examples** tab contains numerous example policy documents that can be applied for different use cases. See the policy documents under that tab or refer to AWS IoT Core policies for additional information.

AWS IoT > Security > Policies > Create policy						
Create policy in						
AWS INT Core policies allow you to manage access to the AWS INT Core data plane operations						
Policy properties						
AWS IOT Core supports named policies so that many identities can reference the same policy document.						
Policy name						
your_policy_name						
A policy name is an alphanumeric string that can also contain period (), comma (), hyphen(-), underscore (), plus sign (+), equal sign (-), and at sign (@) characters, but no spaces.						
rags - optional						
Policy statements Policy examples						
Policy document use	Bu	ilder ISON				
An AWS fold policy contains one or more policy statements. Each policy statement contains actions, resources, and an effect that grants or denies the actions by the resources.	54					
Policy effect Policy action Policy resource						
Allow • • •	Remove					
Add new statement						
		Cancel Create				

- 13. Select **Create** to create the policy.
- 14. On the **Set device permissions** page, refresh your browser (select the icon that looks like a circular arrow pointing to its own starting point). Your policy should now show up in the list under **Policies**. Choose the checkbox in front of your policy, then select **Next**.

AWS IoT > Connect > Connect n Step 1 Describe provisioning template	Any devices > Create provisioning template > Provisioning by unique device certificates Set device permissions Info AWS IoT policies authorize devices to access AWS IoT resources such as other thing resources, MQTT topics, and Device
Step 2 Set provisioning actions	Policies (1/3) Info
Step 3 Set device permissions Step 4 Review and create	Q Find policies < 1 > O
	Policy name ARN your_policy_name Image: arrn:aws:iot:us-west-2: image: policy/your_policy_name
	Cancel Previous Next

15. Review the information you have entered to make sure it is correct. In particular, make sure that the CA certificate shown under **CA certificate configuration** has the same CA certificate ID returned by the AWS CLI command that you ran in a previous step. You can edit a particular section if the information is incorrect. After you verify the information you entered, scroll to the bottom and select **Create template**.

Step 2: Set provisioning actions						
Automatic thing creation - optional						
Automatically create a thing resource when provisioning a device On Thing name prefix - Attributes - <i>optional</i>	Thing groups -	Thing type - Billing group -				
Key	/2/14	Тире				
Ney	auc	1 Abc				
No attributes have been configured.						
Step 3: Set device permissions						

Policies				
Policy name your_policy_name	Policy action *	Policy effect Allow		
	Cancel	Previous Create template		

16. Your JITP template is now created and ready to be applied whenever an ExpressLink module issues a connect request to your AWS account's IoT Core endpoint during onboarding-by-claim.

Appendix D - Glossary

This is a glossary, containing all the terms used in this guide.

ExpressLink modules

Hardware connectivity modules that enable easy AWS cloud connectivity and implement strict and AWS-mandated security requirements for device-to-cloud connections.

ExpressLink devices

OEM products that embed an ExpressLink module and use the module for AWS cloud connectivity.

OEM AWS account

An AWS account that OEMs use to manage their IoT devices.

ExpressLink AWS staging account/endpoint

An AWS-managed account that provides:

- out-of-the-box Quick Connect functionality.
- a staging area for non-onboarded ExpressLink devices waiting to be claimed and onboarded.

onboarding

The registering of an OEM's ExpressLink device inside the OEM AWS Account; ExpressLink provides 5 ways to do this.

onboarding-by-claim

One of the 5 onboarding methods provided by ExpressLink; it streamlines onboarding and eliminates manual setup.

claim-thing

A virtual device in the staging account (registry) whose sole purpose is to publish configuration messages.

registration portal

A web or mobile application that allows the end-user to register an (IoT) product.

Onboarding functionality

Any piece of code that is able to:

- connect to the ExpressLink AWS staging account as the claim-thing.
- publish the MQTT endpoint change message to the MQTT control topic in the staging account.