Data transfer to Amazon Web Services (AWS) with an S7-1x00

Mqtt

https://support.industry.siemens.com/cs/ww/en/view/109772284

Siemens Industry Online Support

Legal information

Table of contents

1 Introduction

1.1 Overview

Digitalization has far-reaching effects on society and the economy. Cloud computing is an important prerequisite for being able to exploit the benefits of digitalization in industry.
This application example describes how you can directly connect a SIMATIC S7 controller (without additional gateway solutions) to a wide range of cloud platforms that support the standardized MQTT protocol – for example Amazon Web Services.

The "LMQTT" library is a suitable solution for implementing the MQTT protocol in a SIMATIC S7 controller.

Note The "LMQTT" library is part of the libraries for communication. You can download the library separately from Siemens Industry Online Support
(see \5\ in chapter 4.3).

Another example of an application of the library can be found under \6\ in chapter 4.3.

1.2 Principle of operation

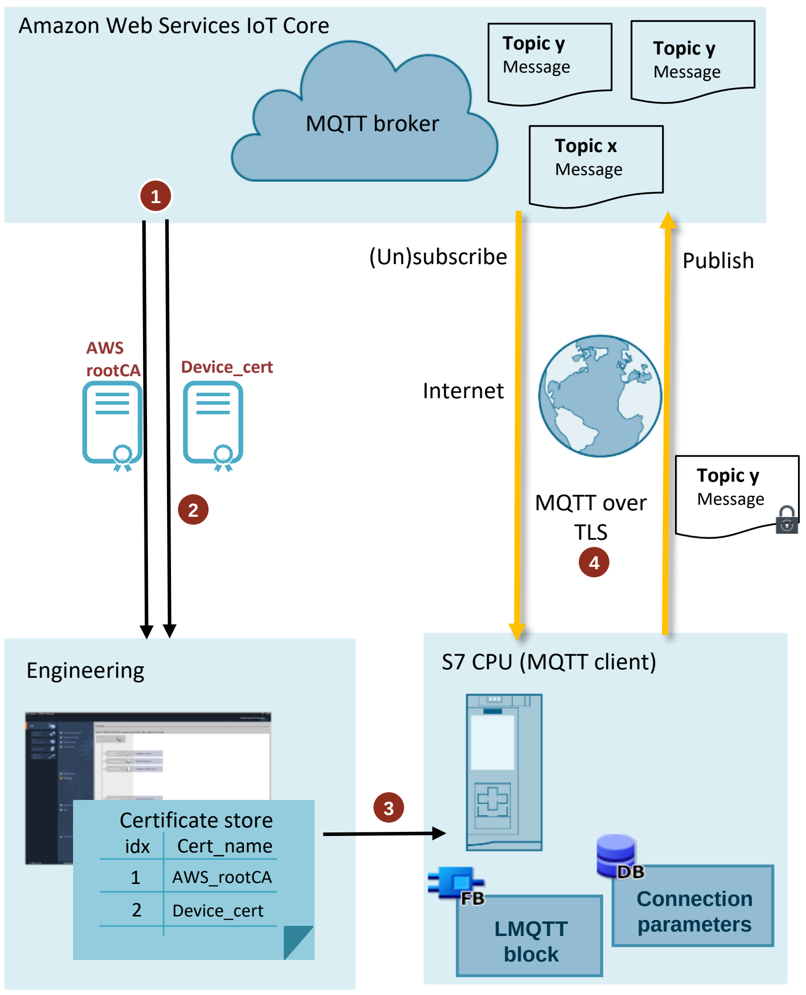
This application example will show you how to connect a SIMATIC S7 controller with Amazon Web Services (AWS) using the "LMQTT" library with IoT Core.

AWS IoT Core offers secure, bidirectional communication between the devices and the AWS cloud. All data traffic to and from AWS IoT is encrypted with Transport Layer Security (TLS). During server and client authentication, your device must authenticate itself with AWS IoT.

This application example uses X.509 certificates for both authentication processes. More information on security in AWS IoT can be found under \3\ in chapter 4.3.

Simplified representation

The following Figure shows the most important relationships between the involved components and the steps required for secure MQTT communication (MQTT over TLS).



Step	Description
1	The device certificate (with the associated keys) is generated and activated. The root CA certificate from AWS is exported.
2	The certificates are imported into TIA Portal. The certificates are now located in the global certificate manager of STEP 7.
3	The certificates are assigned to the S7 CPU.
4	The "LMQTT_Client" function block is responsible for the following roles: • Publisher: Send MQTT messages to broker • Subscriber: Subscribe to MQTT messages or unsubscribe The MQTT messages are encrypted via a secure connection (MQTT over TLS).

Components used

The following hardware and software components were used to create this application example:

- S7-1500 controller
  (here: Item number 6ES7516-3AN02-0AB0)
- "LMQTT" library (contained in the user program)
- Internet access (router, DNS server)
- Engineering station (PG) with TIA Portal V17 or higher
- "Amazon Web Services" account with the IoT Core service set up and licensed. You can find additional information at https://console.aws.amazon.com/

Note You can also use a different SIMATIC S7 controller. For secure MQTT communication over TLS you will need one of the following components:

• S7-1500 CPU with firmware V2.0 or higher

• S7-1200 CPU with firmware V4.4 or higher

This application example consists of the following components:

Component	File name	Note
Documentation	10972284_Datatransfer_AWS_S7-1500_DOKU_en.pdf	This document
Project	109772284_Datatransfer_AWS_CODE.zip	TIA Portal V17 project

2 Engineering

2.1 Certificate manager

Certificate store in TIA Portal

TIA Portal manages two certificate stores:

- Local CPU certificate manager
- Global certificate manager

You can activate the global certificate store in TIA Portal with the option "Use global security settings for certificate manager".

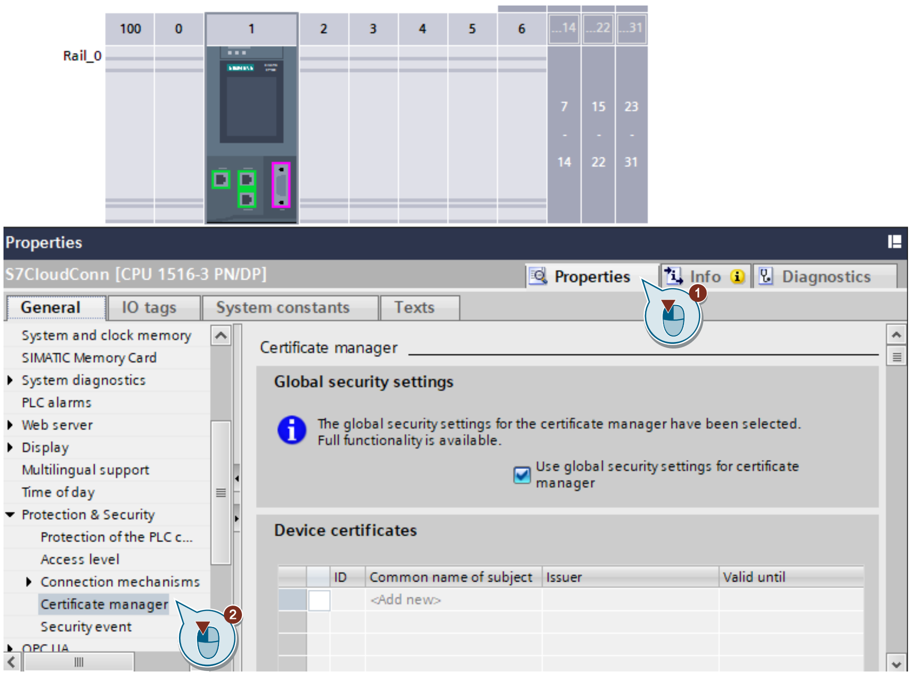
This option decides whether you have access to all certificates in the project or not and whether you can have the device certificates signed by the CA certificates of the certification authority or not.

- If you do not use the certificate manager in the global security settings, then you only have access to the local certificate store of the CPU. You have no access, for example, to imported certificates or root certificates. Without these certificates only a limited functionality is available. For example, you can only generate self-signed certificates.
- If you use the certificate manager in the global security settings, you have access to the global, project-wide certificate store. For example, you can assign imported certificates to the CPU or create device certificates issued and signed by the project's certification authority. TIA Portal provides two root certificates (CA certificates) for the entire project.

Each certificate in the certificate store is assigned an ID that can be used to reference the certificate in the program blocks.

Local CPU certificate manager

You can find the local CPU certificate manager in the CPU properties in the section "Protection & Security > Certificate manager".



Certificate manager

Global security settings

The global security settings for the certificate manager have been selected.
Full functionality is available.

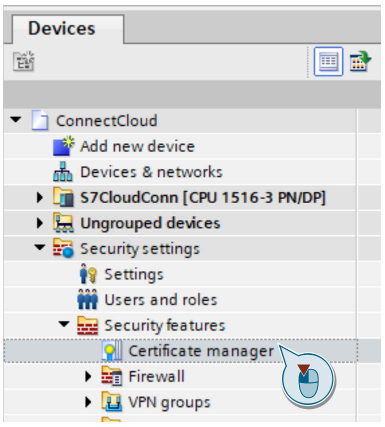
☑ Use global security settings for certificate manager

Device certificates

	ID	Common name of subject	Issuer	Valid until
		<Add new>		

Global certificate manager

If you are signed in to the security settings of your TIA Portal project and have activated the global certificate manager, then you can find the global certificate manager in the project tree of your TIA Portal project under
"Security settings > Security features > Certificate manager".



2.2 Preparing the environment

2.2.1 IP addresses and hardware setup

IP addresses

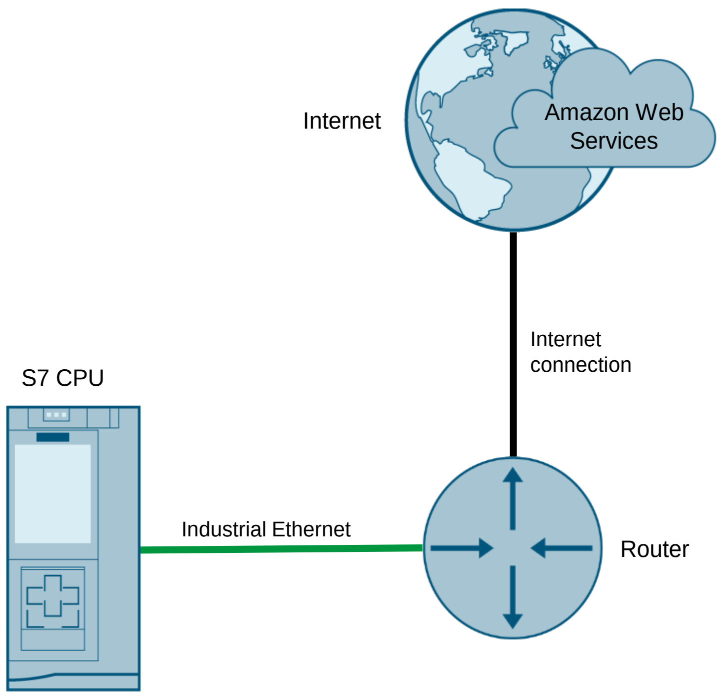
In the TIA Portal project for this application example, the CPU has the following IP addresses:

Table 2-1

Component	IP address	Subnet mask
Ethernet interface	192.168.168.15	255.255.255.0
Router	192.168.168.1	255.255.255.0
DNS server	192.168.168.1	255.255.255.0

Hardware setup

Interconnect the components as shown in the Figure below.



2.2.2 TIA Portal project

Create project

Follow these steps:

1. Open your TIA Portal project or create a new project with your hardware components.
2. When you add the CPU, follow the instructions from the Security Wizard and adjust the security features to suit your needs.
3. Assign the CPU the necessary IP addresses (see Table 2-1).

Set clock time

Because a certificate always has a time period over which it is valid, the clock time of the S7 CPU that wants to encrypt with this certificate must also be in this time period.

With a brand new S7 CPU or after an overall reset of the S7 CPU, the internal clock is set to a default value that lies outside the certificate runtime. The certificate is then marked as invalid.

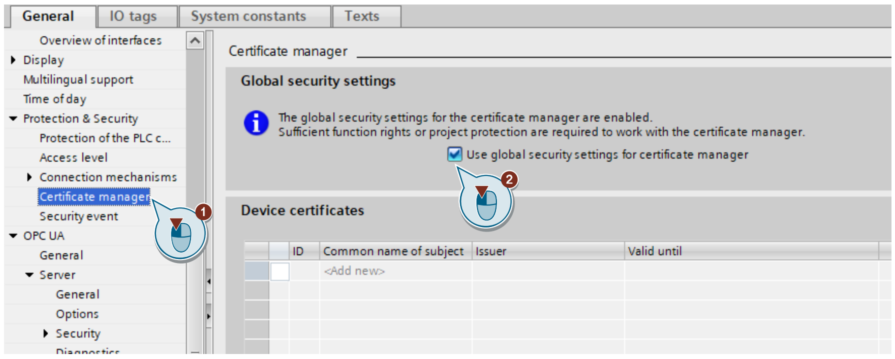
You set the time manually with the Online and Diagnostics view or set up time synchronization.

Enable global certificate manager

To import and use third-party certificates, you must enable the global certificate manager and sign in with the security settings.

Follow these steps:

1. Switch to the menu "Protection & Security > Certificate manager".
   Activate the function "Use global security settings for certificate manager".

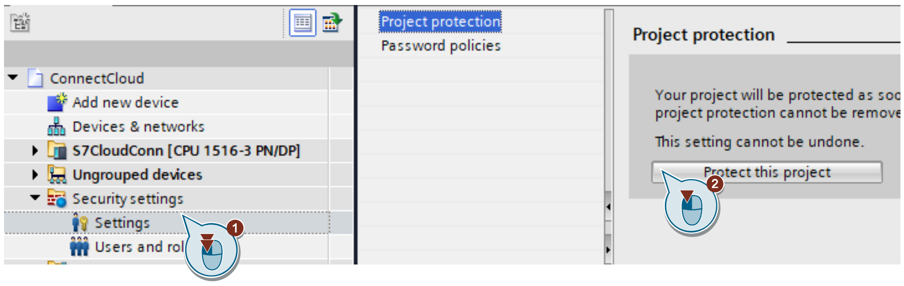


The global security settings for the certificate manager are enabled.
Sufficient function rights or project protection are required to work with the certificate manager.

[✓] Use global security settings for certificate manager

	ID	Common name of subject	Issuer	Valid until
		<Add new>		

2. Double-click "Settings" in the project tree under "Security settings" and click the button "Protect this project".

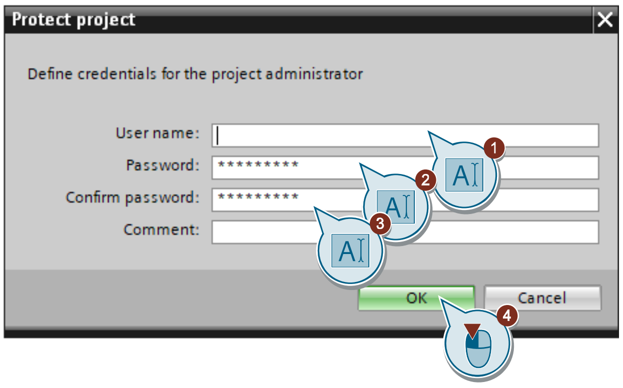


Your project will be protected as soo
project protection cannot be remove

This setting cannot be undone.

The dialog "Project Protection" is shown.

3. In the dialog that appears, define the login information and confirm your entries with "OK".


Protect project
Define credentials for the project administrator
User name:
Password: *********
Confirm password: *********
Comment:
OK
Cancel
1
2
3
4

Note

The login data for the TIA Portal project in this application example are:

User name: ConnectCloud

Password: CloudUser123!

2.3 Using the "LMQTT" library

In this chapter, you will integrate the "LMQTT" library and a global data block into your TIA Portal project.

Note

This application example provides you with a TIA Portal project. The "LMQTT" library has already been added to this TIA Portal project and parameterized.

The login data for the TIA Portal project in this application example are:

User name: ConnectCloud
Password: CloudUser123!

2.3.1 Integrate library blocks

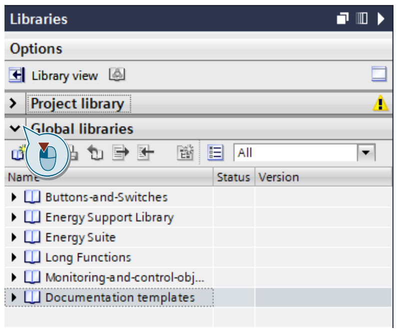
Download library

Download the "Library for communication". You can find the library in Siemens Industry Online Support (see \5\ in chapter 4.3).

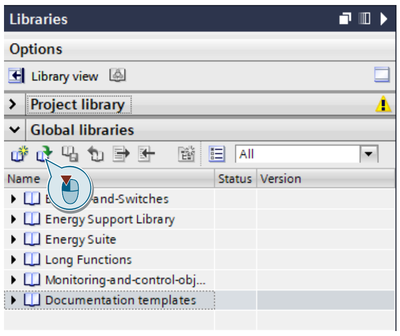
Open "Library for communication"

The "LMQTT" library is part of the "Library for communication". Proceed as follows to open the "Library for communication":

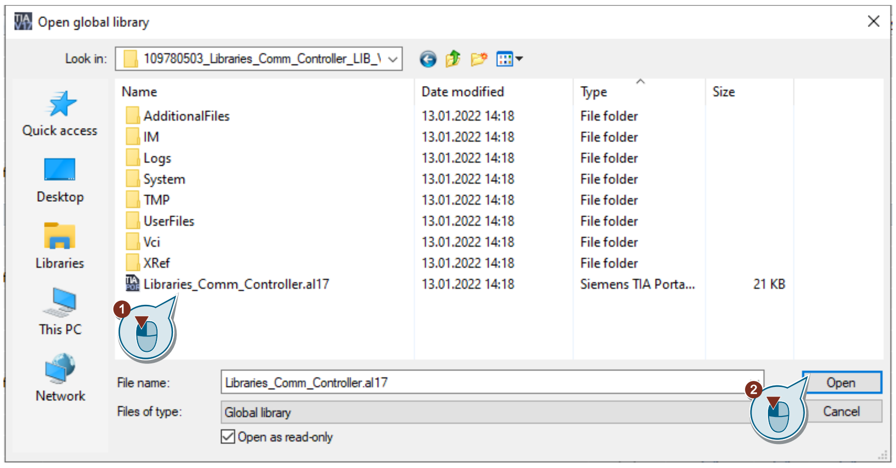
1. In your TIA Portal project, click the "Libraries" tab and open the "Global libraries" palette.



2. Click the "Open global library" button. The "Open global library" dialog will open.



3. Navigate to the directory with the global library and select the global library "Libraries_Comm_Controller.al17". Confirm your selection by clicking the "Open" button.

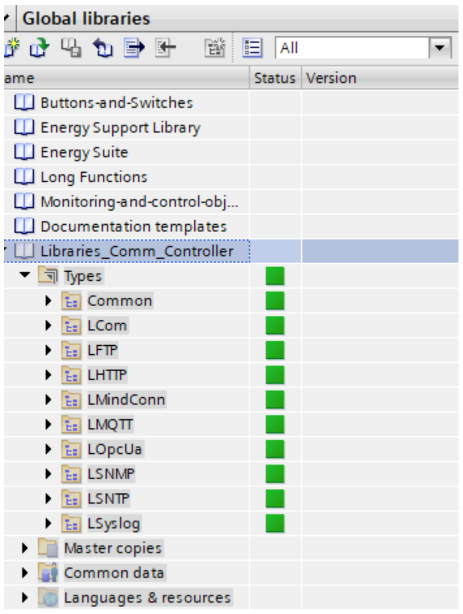


Name	Date modified	Type	Size
AdditionalFiles	13.01.2022 14:18	File folder	
IM	13.01.2022 14:18	File folder	
Logs	13.01.2022 14:18	File folder	
System	13.01.2022 14:18	File folder	
TMP	13.01.2022 14:18	File folder	
UserFiles	13.01.2022 14:18	File folder	
Vci	13.01.2022 14:18	File folder	
XRef	13.01.2022 14:18	File folder	
Libraries_Comm_Controller.al17	13.01.2022 14:18	Siemens TIA Porta...	21 KB

File name: Libraries_Comm_Controller.al17
Files of type: Global library
☑ Open as read-only

Result

The "Libraries_Comm_Controller" library now appears under the "Global libraries" palette.



ame	Status	Version
Buttons-and-Switches		
Energy Support Library		
Energy Suite		
Long Functions		
Monitoring-and-control-obj...		
Documentation templates		
Libraries_Comm_Controller		
Types	🟩	
Common	🟩	
LCom	🟩	
LFTP	🟩	
LHTTP	🟩	
LMindConn	🟩	
LMQTT	🟩	
LOpcUa	🟩	
LSNMP	🟩	
LSNTP	🟩	
LSyslog	🟩	
Master copies		
Common data		
Languages & resources		

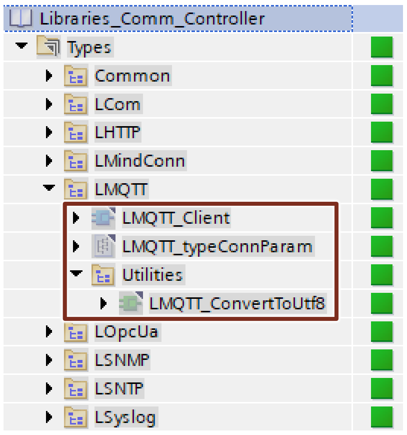
Copy the "LMQTT" library

The "LMQTT" library consists of the following blocks:

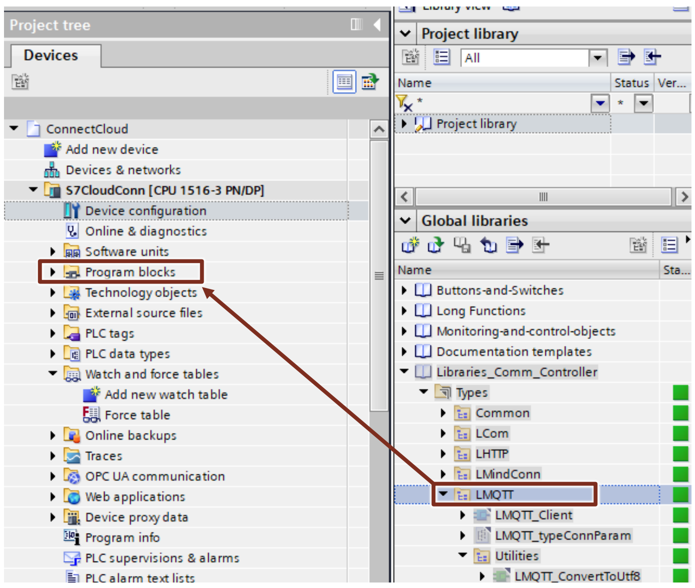
- Function block "LMQTT_Client"
- Data type "LMQTT_typeConnParam"
- System block "LMQTT_ConvertToUtf8"

To copy the "LMQTT" library into your user program, follow these steps:

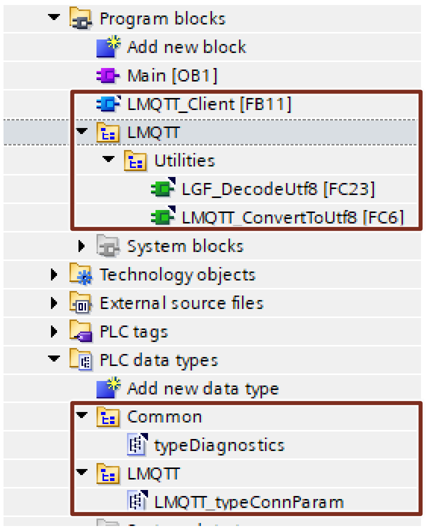
1. Open the folder "Types > LMQTT" in the library "Libraries_Comm_Controller".



2. Drag and drop the "LMQTT" folder into the "Program blocks" folder of your device, e.g. the S7-1500 CPU.



3. The data types required by the function block will be automatically added to the "PLC data types" folder of your CPU.


Program blocks
Add new block
Main [OB1]
LMQTT_Client [FB11]
LMQTT
Utilities
LGF_DecodeUtf8 [FC23]
LMQTT_ConvertToUtf8 [FC6]
System blocks
Technology objects
External source files
PLC tags
PLC data types
Add new data type
Common
typeDiagnostics
LMQTT
LMQTT_typeConnParam

2.3.2 Create global data block

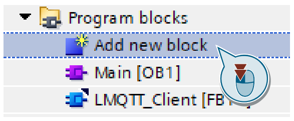
In this chapter you will create a global data block with the following parameters:

- Control and output parameters
- Connection parameters
- MQTT parameters
- Publish parameters
- Subscribe parameters

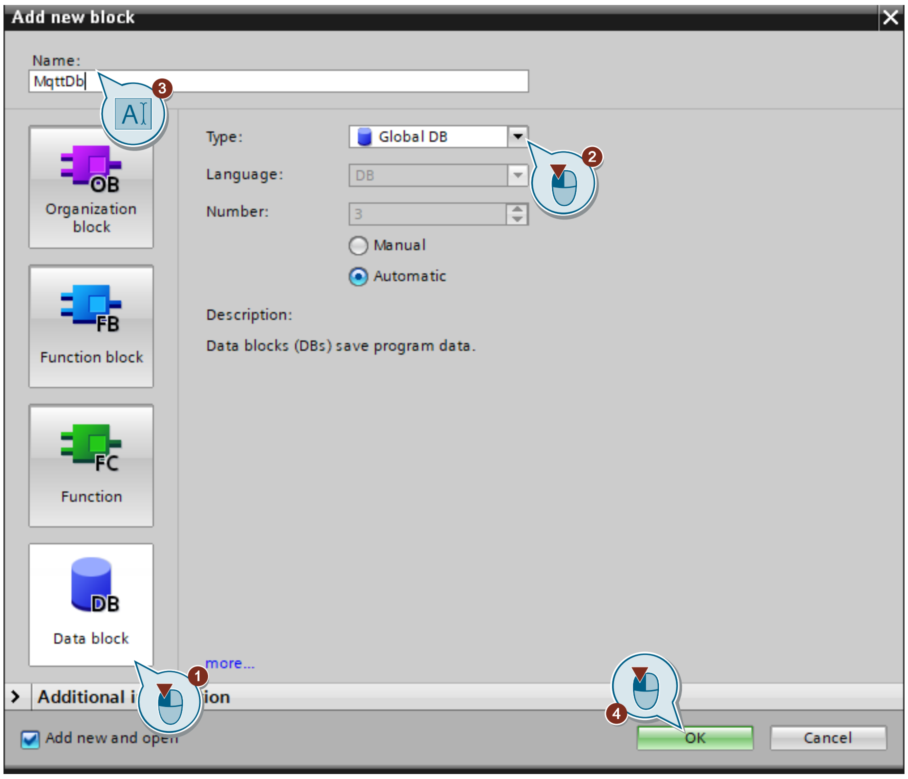
Create data block

Proceed as follows to create a new global data block:

1. In the project tree, navigate to the device folder of the S7 CPU.
2. Open the "Program blocks" folder and double-click the "Add new block" command. The corresponding dialog box opens.


Program blocks
Add new block
Main [OB1]
LMQTT_Client [FB

3. Create a new Global DB and assign a block name, e.g. "MqttDb". Acknowledge the dialog with "OK".


Add new block

Name: MqttDb

Type: Global DB

Language: DB

Number: 3

Manual

Automatic

Description:

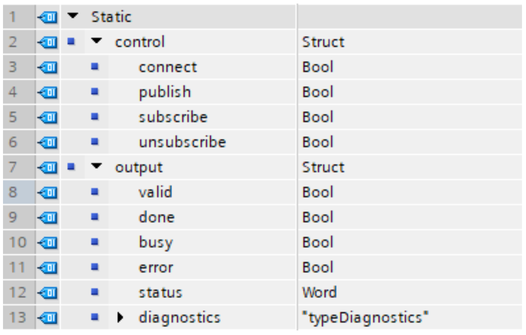
Data blocks (DBs) save program data.

Result

The new data block appears in the program folder of the CPU.

Control and output parameters

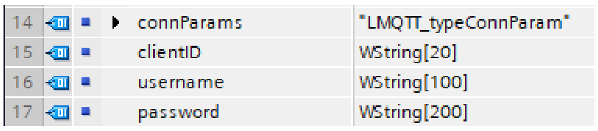
Open the data block. Define the following control and output tags in the data block. Double-click "<Add new>" in an empty row to do this.



1	Static	
2	control	Struct
3	connect	Bool
4	publish	Bool
5	subscribe	Bool
6	unsubscribe	Bool
7	output	Struct
8	valid	Bool
9	done	Bool
10	busy	Bool
11	error	Bool
12	status	Word
13	diagnostics	"typeDiagnostics"

Connection parameters

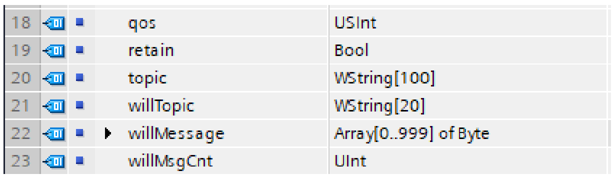
Define the following connection tags in the data block. Double-click "<Add new>" in an empty row to do this.



14	connParams	"LMQTT_typeConnParam"
15	clientID	WString[20]
16	username	WString[100]
17	password	WString[200]

MQTT parameters

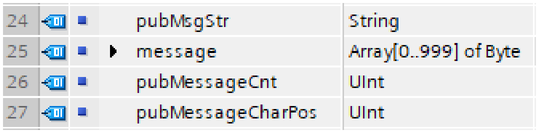
Define the following MQTT tags in the data block. Double-click "<Add new>" in an empty row to do this.



18	qos	USInt
19	retain	Bool
20	topic	WString[100]
21	willTopic	WString[20]
22	willMessage	Array[0..999] of Byte
23	willMsgCnt	UInt

Publish parameters

Define the following publish tags in the data block. Double-click "<Add new>" in an empty row to do this.

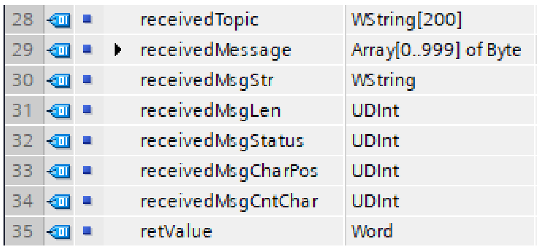


24	pubMsgStr	String
25	message	Array[0..999] of Byte
26	pubMessageCnt	UInt
27	pubMessageCharPos	UInt

Note You can also change the size of the "message" byte array as needed.

Subscribe parameters

Define the following subscribe tags in the data block. Double-click "<Add new>" in an empty row to do this.

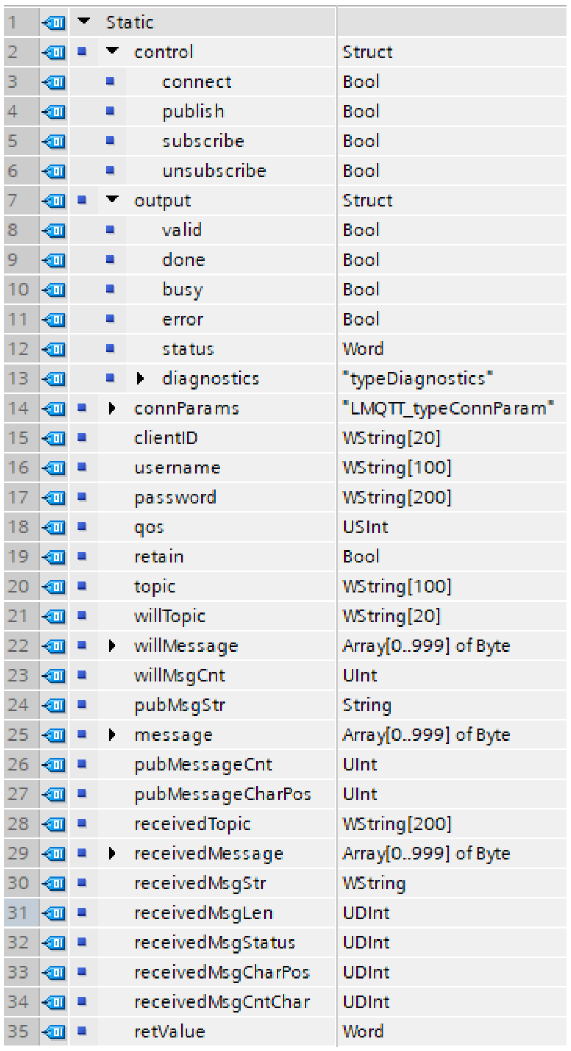


28	receivedTopic	WString[200]
29	receivedMessage	Array[0..999] of Byte
30	receivedMsgStr	WString
31	receivedMsgLen	UDInt
32	receivedMsgStatus	UDInt
33	receivedMsgCharPos	UDInt
34	receivedMsgCntChar	UDInt
35	retValue	Word

Note You can also change the size of the "receivedMessage" byte array as needed.

Result

The data block with all the necessary tags has been declared.



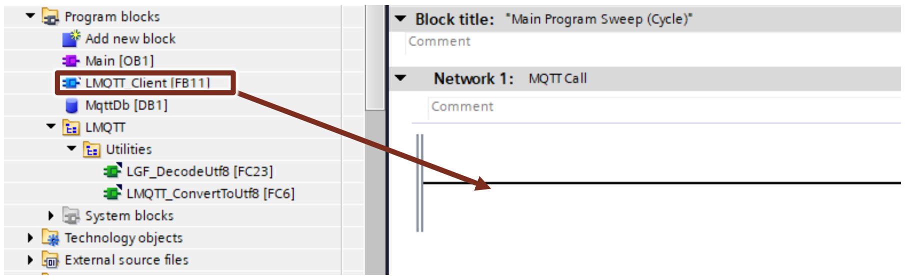
1	Static	
2	control	Struct
3	connect	Bool
4	publish	Bool
5	subscribe	Bool
6	unsubscribe	Bool
7	output	Struct
8	valid	Bool
9	done	Bool
10	busy	Bool
11	error	Bool
12	status	Word
13	diagnostics	"typeDiagnostics"
14	connParams	"LMQTT_typeConnParam"
15	clientID	WString[20]
16	username	WString[100]
17	password	WString[200]
18	qos	USInt
19	retain	Bool
20	topic	WString[100]
21	willTopic	WString[20]
22	willMessage	Array[0..999] of Byte
23	willMsgCnt	UInt
24	pubMsgStr	String
25	message	Array[0..999] of Byte
26	pubMessageCnt	UInt
27	pubMessageCharPos	UInt
28	receivedTopic	WString[200]
29	receivedMessage	Array[0..999] of Byte
30	receivedMsgStr	WString
31	receivedMsgLen	UDInt
32	receivedMsgStatus	UDInt
33	receivedMsgCharPos	UDInt
34	receivedMsgCntChar	UDInt
35	retValue	Word

2.3.3 Call "LMQTT_Client" in the user program and connect it

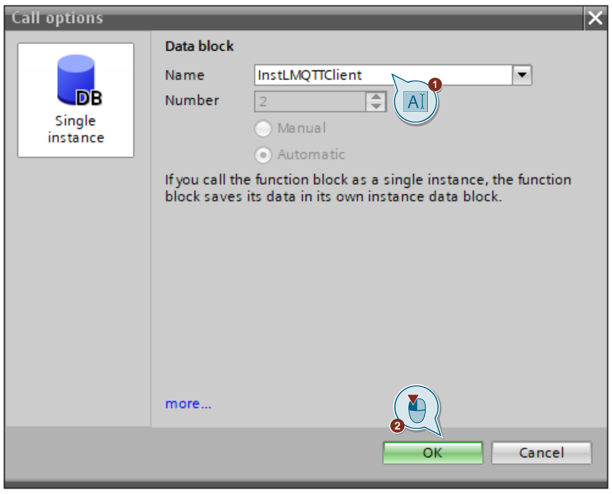
Call "LMQTT_Client" block

Proceed as follows to call the "LMQTT_Client" block in the user program:

1. In the Project tree, open the folder "Program blocks" of your CPU.
2. Double-click the block "Main [OB1]" to open the associated program editor.
3. Drag & drop the block "LMQTT_Client" from the project tree to any OB1 network.



4. The corresponding instance DB will be created. Change the block name if necessary. Confirm the dialog with "OK".



Name: InstLMQTTClient

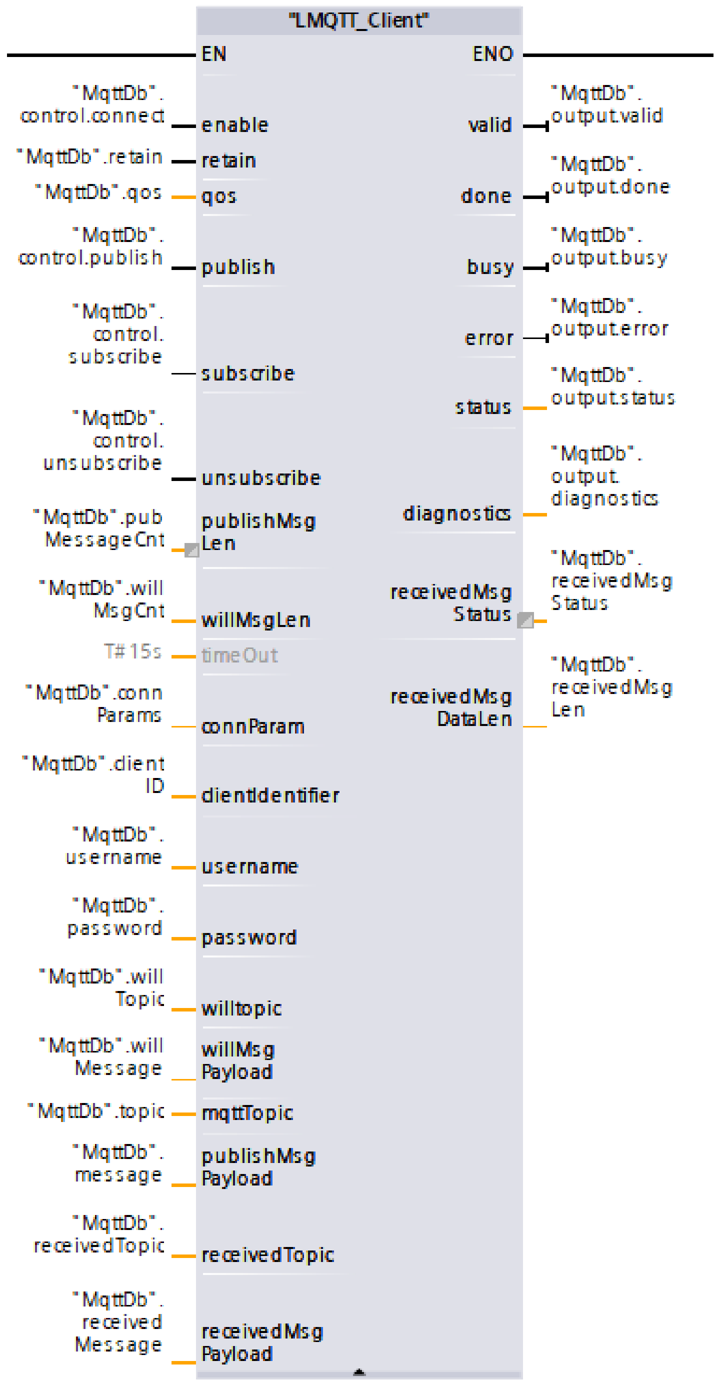
Number: 2

Automatic

If you call the function block as a single instance, the function block saves its data in its own instance data block.

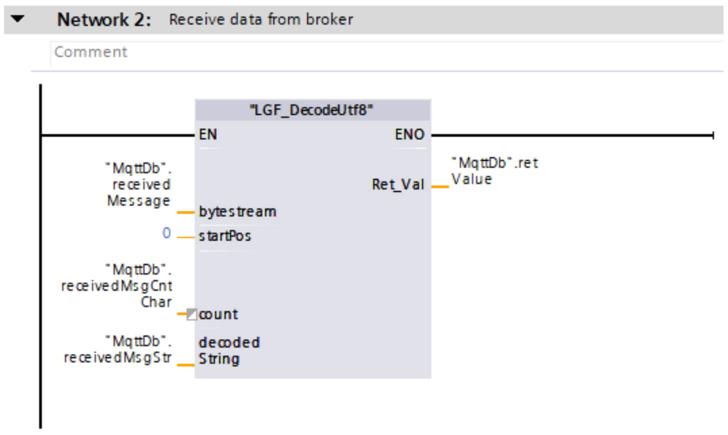
Connect block

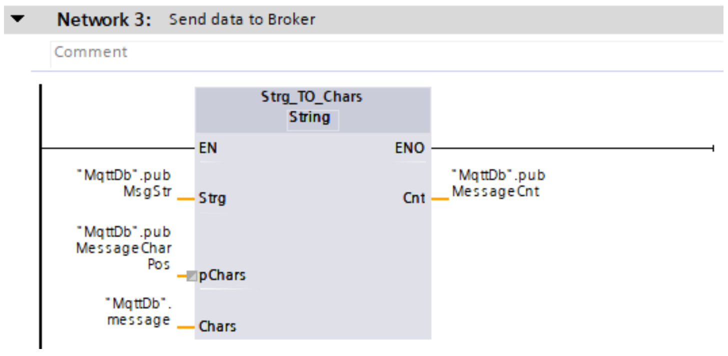
Connect the inputs and outputs of the "LMQTT_Client" FB with the tags that you created in the "MqttDb" data block (see chapter 2.3.2).


"LMQTT_Client"
EN
ENO
"MqttDb".control.connect — enable
valid — "MqttDb".output.valid
"MqttDb".retain — retain
"MqttDb".qos — qos
done — "MqttDb".output.done
"MqttDb".control.publish — publish
busy — "MqttDb".output.busy
error — "MqttDb".output.error
"MqttDb".control.subscribe — subscribe
status — "MqttDb".output.status
"MqttDb".control.unsubscribe — unsubscribe
diagnostics — "MqttDb".output.diagnostics
"MqttDb".pubMessageCnt — publishMsgLen
receivedMsgStatus — "MqttDb".receivedMsgStatus
"MqttDb".willMsgCnt — willMsgLen
T#15s — timeOut
receivedMsgDataLen — "MqttDb".receivedMsgLen
"MqttDb".connParams — connParam
"MqttDb".clientID — clientIdentifier
"MqttDb".username — username
"MqttDb".password — password
"MqttDb".willTopic — willtopic
"MqttDb".willMessage — willMsgPayload
"MqttDb".topic — mqttTopic
"MqttDb".message — publishMsgPayload
"MqttDb".receivedTopic — receivedTopic
"MqttDb".receivedMessage — receivedMsgPayload

Insert additional networks

To convert the messages to and from the broker as a string, add the following networks to the "Main [OB1]" building block.





2.4 Setting up AWS IoT

The following chapter describes the steps required to establish a connection in AWS IoT between the S7 CPU and the AWS cloud and send data.

An "Amazon Web Services" account with the IoT Core service set up and licensed is a prerequisite.

To connect a device with AWS IoT, the following steps are necessary:

- Create IoT object
- Configure device certificate
- Create and attach AWS IoT policy

Note This application example does not consider automatic saving of incoming values. If you wish to save or process data, program this additional functionality in the user program of the application example.

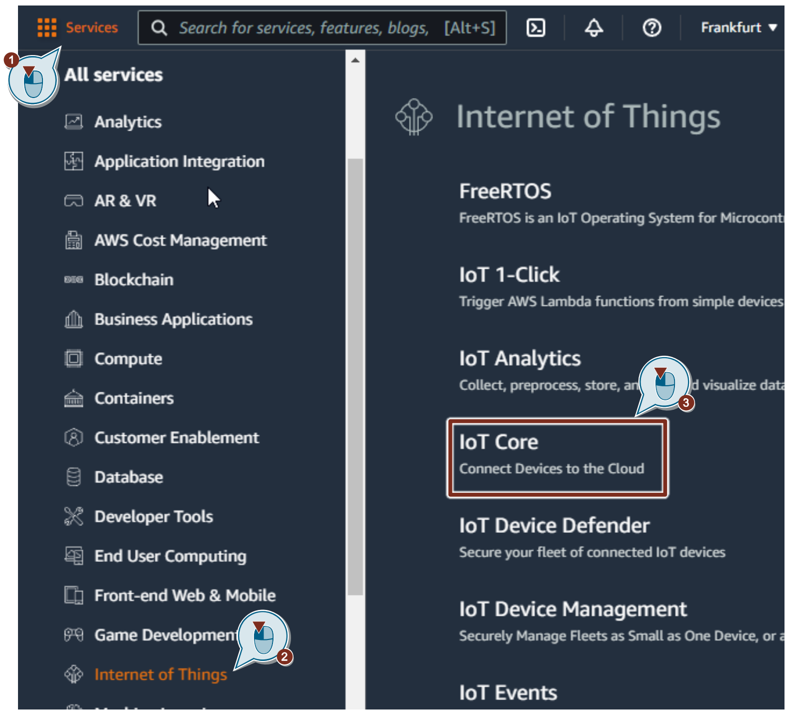
2.4.1 Create AWS IoT object

An "object" is the digital representation of a physical object (in this case, the S7 CPU). It contains static metadata about the device.

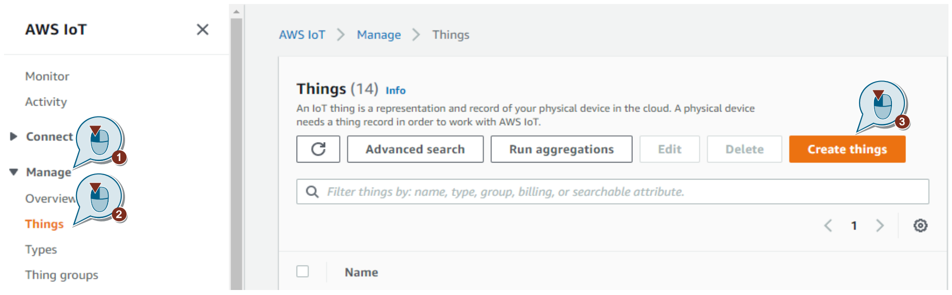
Proceed as follows to create an IoT object:

1. Open the AWS cloud service with the link https://console.aws.amazon.com and sign in with your login credentials.

2. Open the "Services" menu in the top menu bar. In the "Internet of Things" menu, select the "IoT Core" service.

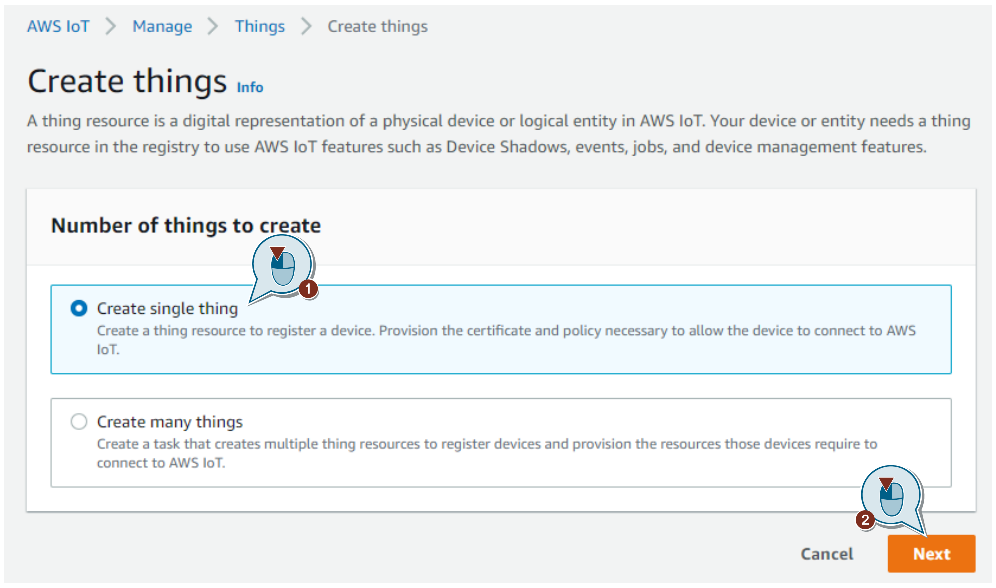


3. In the left pane, navigate to "AWS IoT > Manage > Things" and select "Create things".


AWS IoT
Monitor
Activity
Connect
Manage
Overview
Things
Types
Thing groups
AWS IoT > Manage > Things
Things (14) Info
An IoT thing is a representation and record of your physical device in the cloud. A physical device needs a thing record in order to work with AWS IoT.
Advanced search
Run aggregations
Edit
Delete
Create things
Filter things by: name, type, group, billing, or searchable attribute.
Name
1
2
3

The "Create things" page appears.

4. Click "Create single thing" and click "Next".



Create things Info

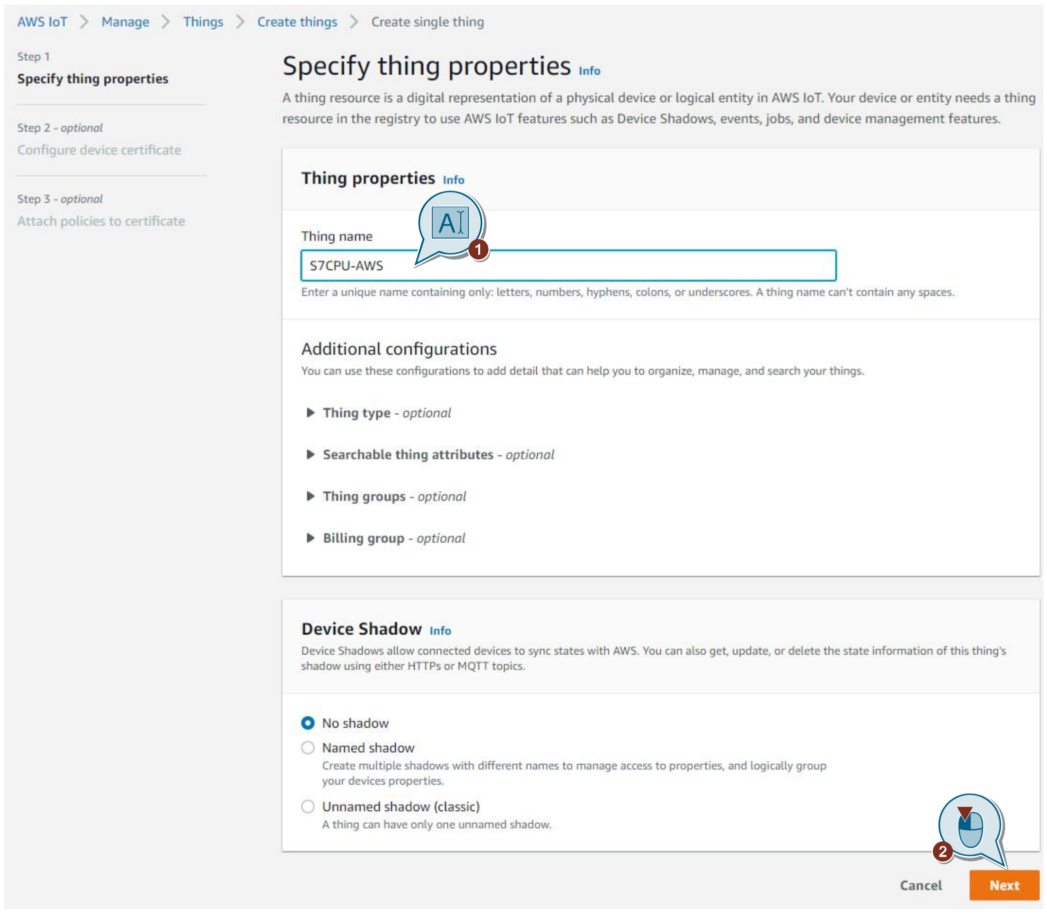
A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

Number of things to create

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

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

5. Give your thing a name.
   You have the option to create additional properties, attributes and groups. No additional properties are defined in this application example.
   Click on "Next".


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

Step 1
Specify thing properties

Step 2 - *optional*
Configure device certificate

Step 3 - *optional*
Attach policies to certificate

Specify thing properties Info

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

Thing properties Info

Thing name
S7CPU-AWS
Enter a unique name containing only: letters, numbers, hyphens, colons, or underscores. A thing name can't contain any spaces.

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

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

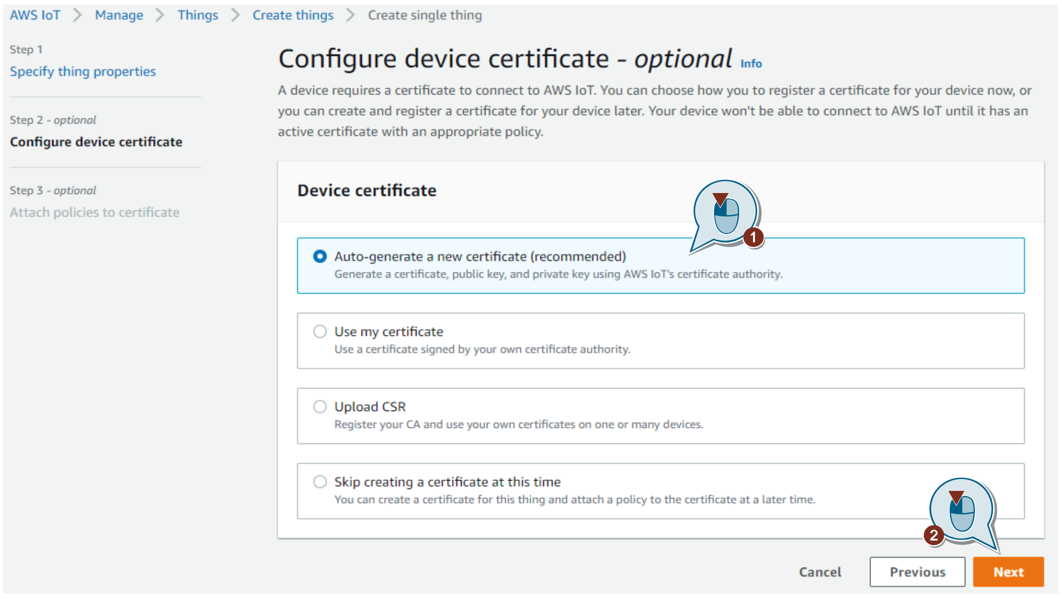
Device Shadow Info
Device Shadows allow connected devices to sync states with AWS. You can also get, update, or delete the state information of this thing's shadow using either HTTPs or MQTT topics.

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

2.4.2 Configure device certificate

Proceed as follows to configure the device certificate:

1. In the next step you will configure the device certificate. Select the item "Auto-generate a new certificate (recommended)".



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

Step 1
Specify thing properties

Step 2 - *optional*
Configure device certificate

Step 3 - *optional*
Attach policies to certificate

Configure device certificate - *optional* Info

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

Device certificate

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

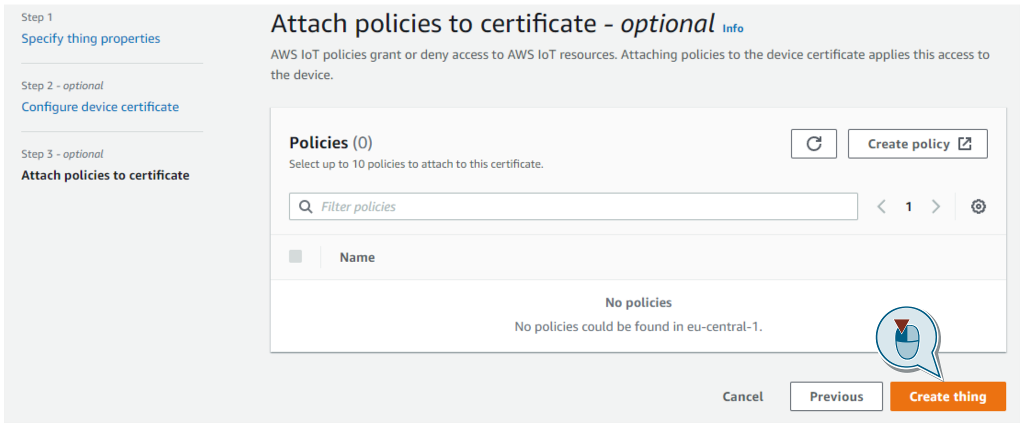
○ Use my certificate
Use a certificate signed by your own certificate authority.

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

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

Note You can also select "Skip creating a certificate at this time" and create the certificate at a later time via "AWS IoT > Secure > Certificates".

2. Click "Create thing" to finish creating the object.



Step 1
Specify thing properties

Step 2 - *optional*
Configure device certificate

Step 3 - *optional*
Attach policies to certificate

Attach policies to certificate - *optional* Info

AWS IoT policies grant or deny access to AWS IoT resources. Attaching policies to the device certificate applies this access to the device.

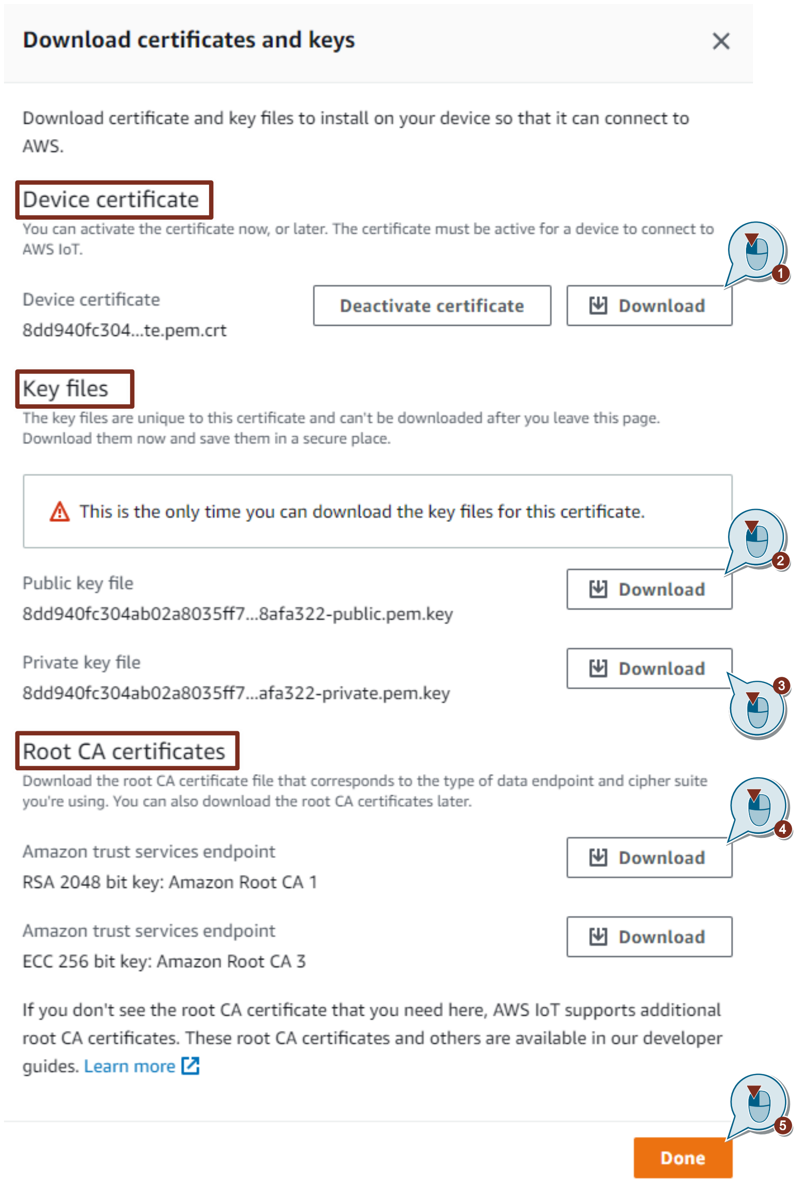
Policies (0)
Select up to 10 policies to attach to this certificate.

No policies
No policies could be found in eu-central-1.

Note If policies already exist for the object that you just created, then attach them in the next step. Chapter 2.4.3 demonstrates how to create a policy and later add an existing object.

3. In the "Download certificates and keys" dialog, you can download the following certificates:
   – the device certificate
   – private and public keys
   – the Amazon root CA certificate ("AmazonRootCA1")

   Save the required keys and certificates to your computer, as they will be imported into your TIA Portal project at a later time.
   For the root CA certificate, select the RSA 2048-bit key.
   Click the "Done" button.



Download certificates and keys

Download certificate and key files to install on your device so that it can connect to AWS.

Device certificate

You can activate the certificate now, or later. The certificate must be active for a device to connect to AWS IoT.

Device certificate
8dd940fc304...te.pem.crt

Key files

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

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

Public key file
8dd940fc304ab02a8035ff7...8afa322-public.pem.key

Private key file
8dd940fc304ab02a8035ff7...afa322-private.pem.key

Root CA certificates

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

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

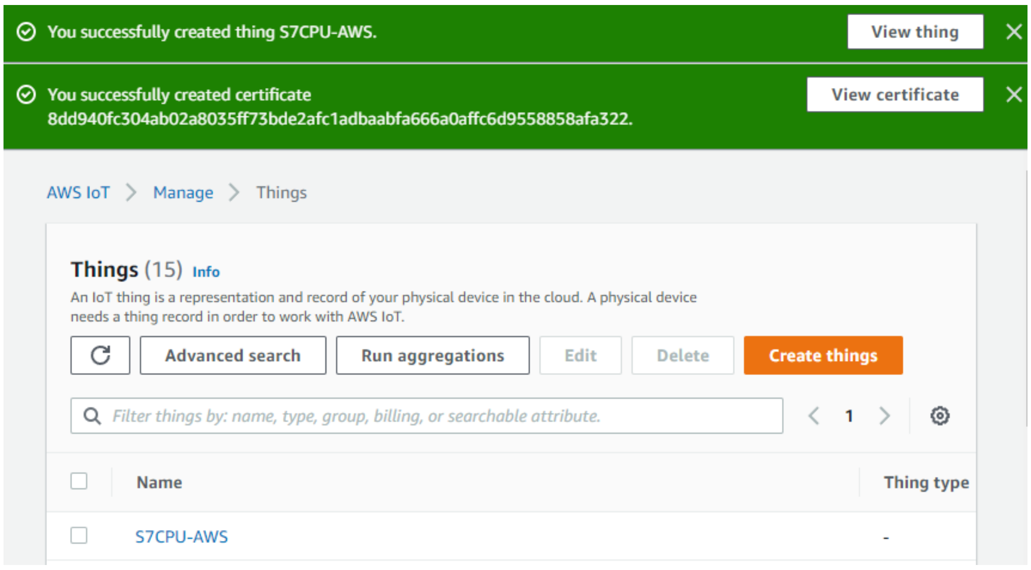
Amazon trust services endpoint
ECC 256 bit key: Amazon Root CA 3

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

CAUTION	**It is not possible to download the key files after you have exited the "Download certificates and key" dialog.** Download them now and save them to a secure location.

Result

The thing has been created successfully.



You successfully created thing S7CPU-AWS.

You successfully created certificate 8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322.

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

Advanced search
Run aggregations
Edit
Delete
Create things
Filter things by: name, type, group, billing, or searchable attribute.
1

Name	Thing type
S7CPU-AWS	-

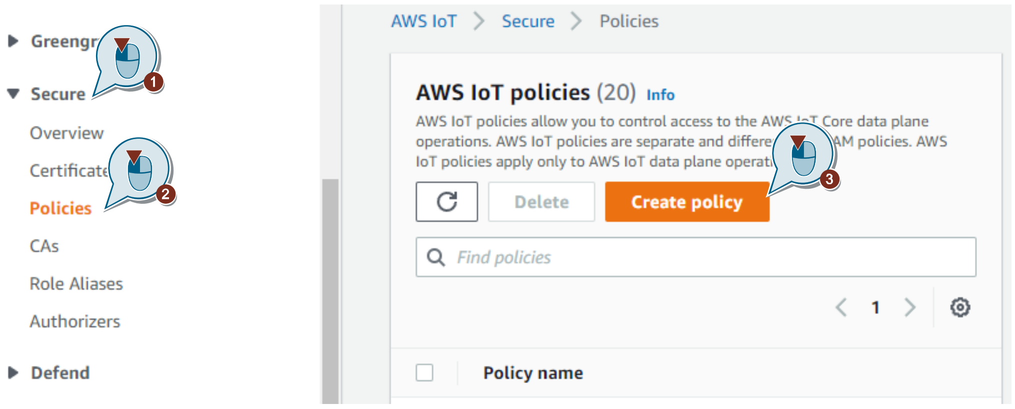
2.4.3 Create and attach AWS IoT policy

The policies define which AWS IoT resources the devices have access to. At least one AWS IoT policy is assigned to each device certificate.

Create IoT AWS policy

In the following, you will create a policy for unrestricted access.

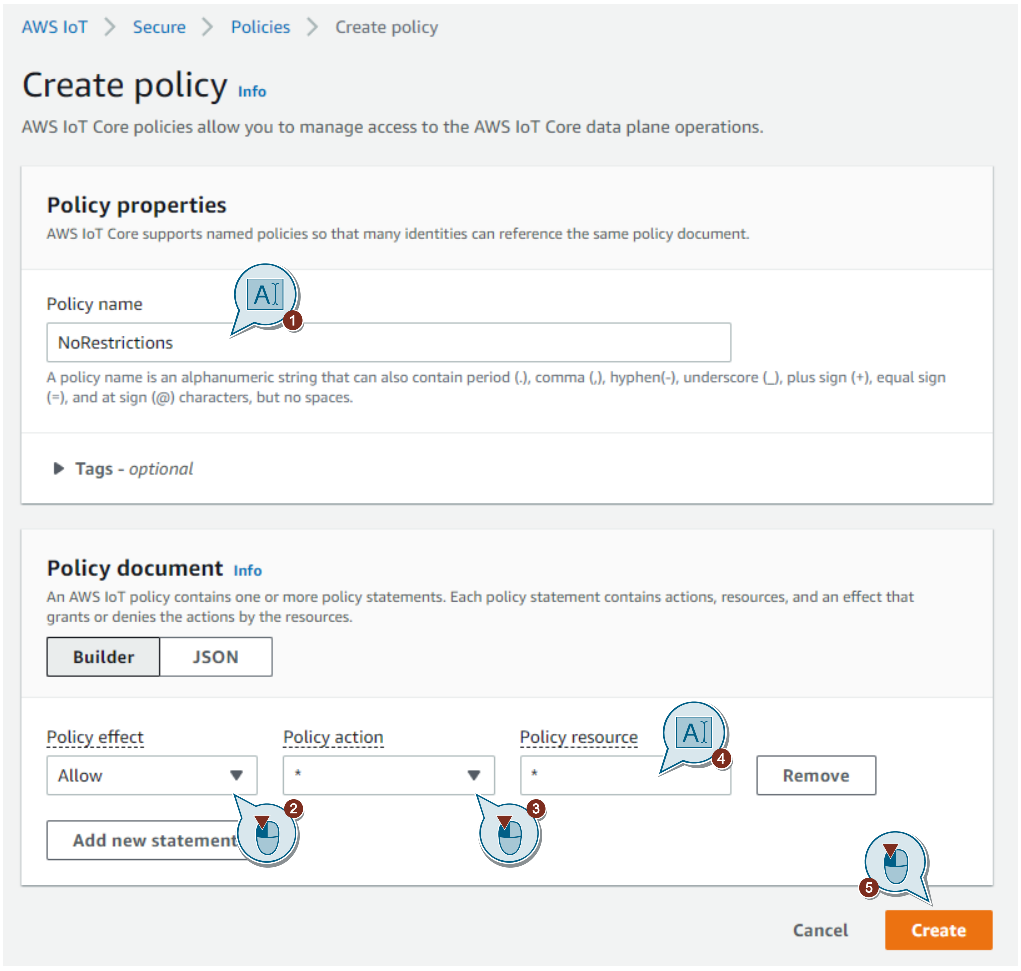
1. Navigate to "AWS IoT > Secure > Policies" and select "Create policy".



AWS IoT policies allow you to control access to the AWS IoT Core data plane operations. AWS IoT policies are separate and differe AM policies. AWS IoT policies apply only to AWS IoT data plane operati

2. Set a name for your policy, in this example "NoRestrictions", and define the following policy instructions for unrestricted access:
   - "Policy effect": "Allow"
   - "Policy action": "*"
   - "Policy resources": "*"

   Click the "Create" button.



AWS IoT > Secure > Policies > Create policy

Create policy Info

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

Policy properties

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

Policy name: NoRestrictions

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.

▶ Tags - *optional*

Policy document Info

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

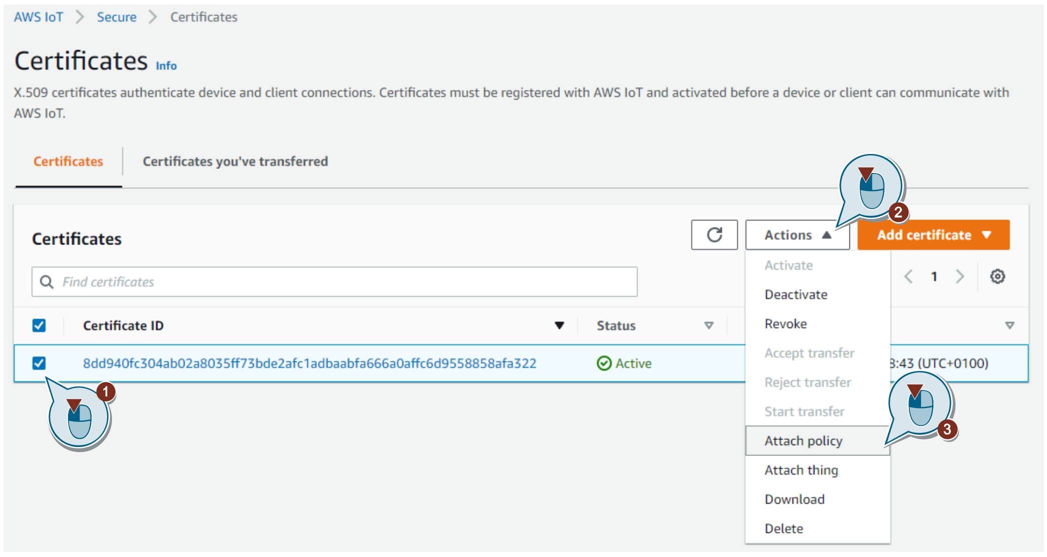
Builder | JSON

Policy effect	Policy action	Policy resource
Allow	*	*

Attach IoT AWS policy

Proceed as follows to attach a policy:

1. Navigate to "Secure > Certificates". Tick the checkbox to select the device certificate. In the "Actions" dropdown menu, select the "Attach policy" option.



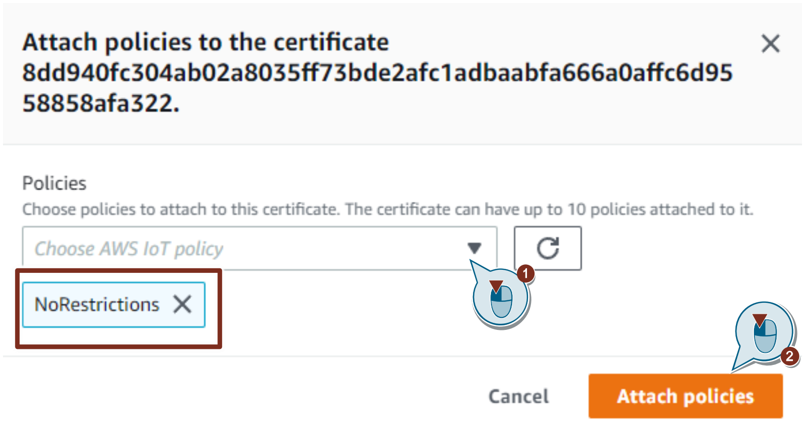
AWS IoT > Secure > Certificates

Certificates Info

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

✓	Certificate ID	Status
✓	8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322	Active

2. To attach policies to the device certificate, select one or more policies in the dialog shown. In this example, select the unrestricted access policy you created earlier from the dropdown menu and click "Attach policies".

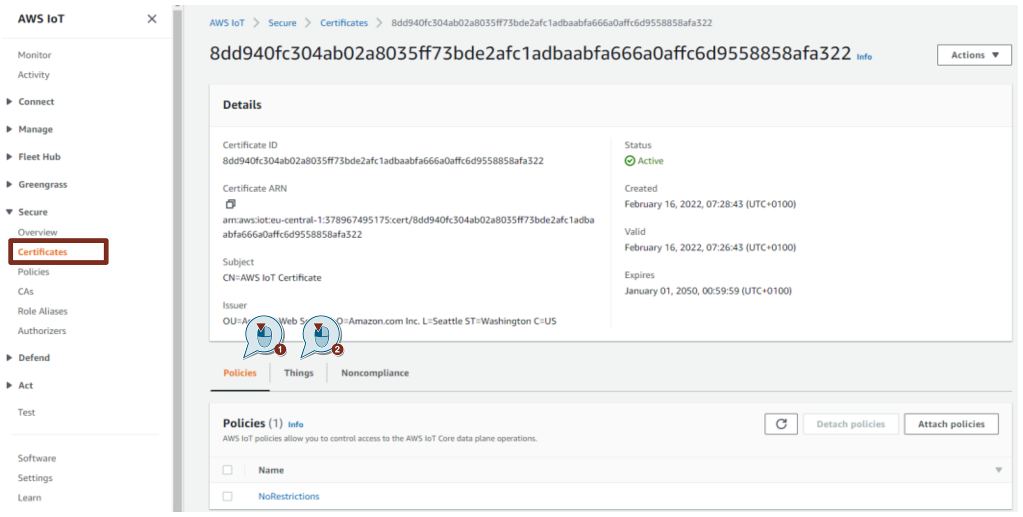


Attach policies to the certificate 8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322.

Policies

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

3. The "Secure > Certificates" menu shows the certificate details, policies and the things linked to the certificate.



8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322 Info

Details

Certificate ID
8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322

Certificate ARN
arn:aws:iot:eu-central-1:378967495175:cert/8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322

Subject
CN=AWS IoT Certificate

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

Status
Active

Created
February 16, 2022, 07:28:43 (UTC+0100)

Valid
February 16, 2022, 07:26:43 (UTC+0100)

Expires
January 01, 2050, 00:59:59 (UTC+0100)

Policies (1) Info
AWS IoT policies allow you to control access to the AWS IoT Core data plane operations.

Name
NoRestrictions

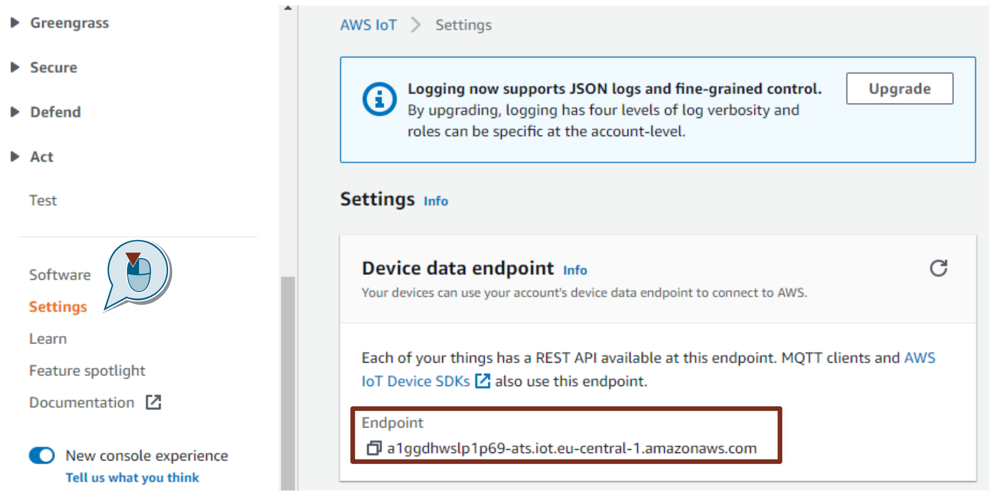
2.4.4 Find device data endpoint

The AWS account has a custom device data endpoint address that devices can connect to for IoT communication. The device data endpoint is needed in the S7 CPU to set up the connection.

Proceed as follows to find the device data endpoint:

1. Navigate to "AWS IoT Hub > Settings".

2. Save the "Endpoint address" in a text file, as the endpoint will be needed for parameter assignment (see chapter 2.6).



Logging now supports JSON logs and fine-grained control.
By upgrading, logging has four levels of log verbosity and roles can be specific at the account-level.

Settings Info

Device data endpoint Info
Your devices can use your account's device data endpoint to connect to AWS.

Each of your things has a REST API available at this endpoint. MQTT clients and AWS IoT Device SDKs also use this endpoint.

Endpoint
a1ggdhwslp1p69-ats.iot.eu-central-1.amazonaws.com

2.5 Configuring safety functions

2.5.1 Prepare device certificate for TIA Portal

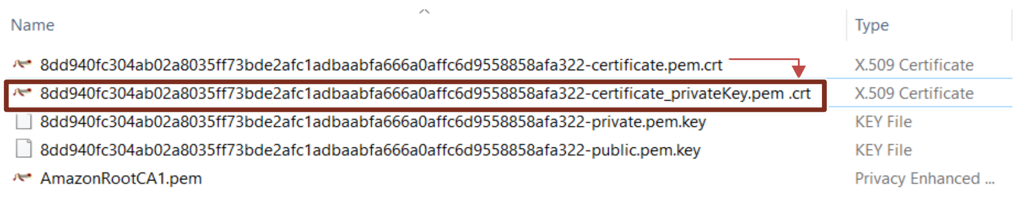
In addition to the server authentication, AWS IoT will also perform an authentication of the MQTT client. Therefore, the device certificate you created in chapter 2.4.2 must be imported into TIA Portal together with its private key.

To do this, it is necessary to combine the device certificate and the private key into a single file. The easiest way to combine a device certificate and a private key is with a text editor. However, the new file will not be encrypted if you use this method. In this application example, this method is used for illustration.

Note To combine the certificates in an encrypted manner, use third-party tools such as X-Certificate and Key management (XCA). Handling of certificates with XCA is explained in chapter 3.

Proceed as follows to combine the device certificate and the private key into one file:

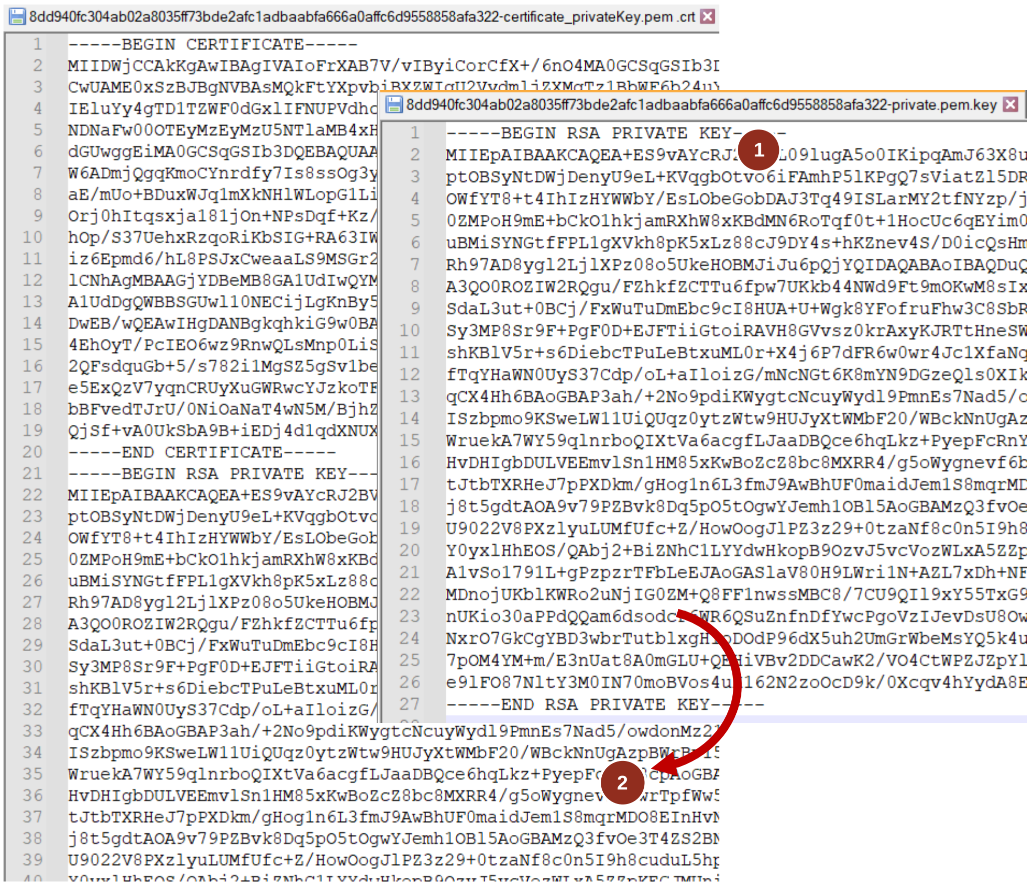
1. Navigate to the direcotry, the certificates are saved (see chapter 2.4.2). Save the device certificate as a copy under a different name.



Name	Type
8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322-certificate.pem.crt	X.509 Certificate
8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322-certificate_privateKey.pem .crt	X.509 Certificate
8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322-private.pem.key	KEY File
8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322-public.pem.key	KEY File
AmazonRootCA1.pem	Privacy Enhanced ...

2. Open the following files with Notepad++ or another text editor.
   – The copy of the device certificate (see Step 1)
   – The private key

   Copy the content of the file with the private key and append it to the contents in the copy of the device certificate. Save the modified copy of the device certificate.



8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322-certificate_privateKey.pem .crt

```
-----BEGIN CERTIFICATE-----
MIIDWjCCAkKgAwIBAgIVAIoFrXAB7V/vIByiCorCfX+/6nO4MA0GCSqGSIb3D
CwUAME0xSzBJBgNVBAsMQkFtYXpvbiBXZWIgU2VydmljZXMgTz1BbWF6b24uY
IEluYy4gTD1TZWF0dGxlIFNUPVdhc
NDNaFw00OTEyMzEyMzU5NTlaMB4xH
dGUwggEiMA0GCSqGSIb3DQEBAQUAA
W6ADmjQgqKmoCYnrdfy7Is8ssOg3y
aE/mUo+BDuxWJq1mXkNHlWLopG1Li
Orj0hItqsxja181jOn+NPsDqf+Kz/
hOp/S37UehxRzqoRiKbSIG+RA63IW
iz6Epmd6/hL8PSJxCweaaLS9MSGr2
lCNhAgMBAAGjYDBeMB8GA1UdIwQYM
A1UdDgQWBBSGUwl10NECijLgKnBy5
DwEB/wQEAwIHgDANBgkqhkiG9w0BA
4EhOyT/PcIEO6wz9RnwQLsMnp0LiS
2QFsdquGb+5/s782i1MgSZ5gSv1be
e5ExQzV7yqnCRUyXuGWRwcYJzkoTF
bBFvedTJrU/0NiOaNaT4wN5M/BjhZ
QjSf+vA0UkSbA9B+iEDj4d1qdXNUX
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY---
MIIEpAIBAAKCAQEA+ES9vAYcRJ2BV
ptOBSyNtDWjDenyU9eL+KVqgbOtvo
OWfYT8+t4IhIzHYWWbY/EsLObeGob
0ZMPoH9mE+bCkO1hkjamRXhW8xKBd
uBMiSYNGtfFPL1gXVkh8pK5xLz88c
Rh97AD8ygl2LjlXPz08o5UkeHOBMJ
A3QO0ROZIW2RQgu/FZhkfZCTTu6fp
SdaL3ut+0BCj/FxWuTuDmEbc9cI8H
Sy3MP8Sr9F+PgF0D+EJFTiiGtoiRA
shKBlV5r+s6DiebcTPuLeBtxuML0r
fTqYHaWN0UyS37Cdp/oL+aIloizG/
qCX4Hh6BAoGBAP3ah/+2No9pdiKWygtcNcuyWydl9PmnEs7Nad5/owdonMz2
ISzbpmo9KSweLW11UiQUqz0ytzWtw9HUJyXtWMbF20/WBckNnUgAzpBW
WruekA7WY59qlnrboQIXtVa6acgfLJaaDBQce6hqLkz+PyepFc[illegible]cpAoGBA
HvDHIgbDULVEEmvlSn1HM85xKwBoZcZ8bc8MXRR4/g5oWygnev[illegible]wrTpfWw5
tJtbTXRHeJ7pPXDkm/gHog1n6L3fmJ9AwBhUF0maidJem1S8mqrMDO8EInHvN
j8t5gdtAOA9v79PZBvk8Dq5pO5tOgwYJemh1OBl5AoGBAMzQ3fvOe3T4ZS2BN
U9022V8PXzlyuLUMfUfc+Z/HowOogJlPZ3z29+0tzaNf8c0n5I9h8cuduL5hp
```

8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322-private.pem.key

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA+ES9vAYcRJ2[illegible]L09lugA5o0IKipqAmJ63X8u
ptOBSyNtDWjDenyU9eL+KVqgbOtvo6iFAmhP5lKPgQ7sViatZl5DR
OWfYT8+t4IhIzHYWWbY/EsLObeGobDAJ3Tq49ISLarMY2tfNYzp/j
0ZMPoH9mE+bCkO1hkjamRXhW8xKBdMN6RoTqf0t+1HocUc6qEYim0
uBMiSYNGtfFPL1gXVkh8pK5xLz88cJ9DY4s+hKZnev4S/D0icQsHm
Rh97AD8ygl2LjlXPz08o5UkeHOBMJiJu6pQjYQIDAQABAoIBAQDuQ
A3QO0ROZIW2RQgu/FZhkfZCTTu6fpw7UKkb44NWd9Ft9mOKwM8sIx
SdaL3ut+0BCj/FxWuTuDmEbc9cI8HUA+U+Wgk8YFofruFhw3C8SbR
Sy3MP8Sr9F+PgF0D+EJFTiiGtoiRAVH8GVvsz0krAxyKJRTtHneSW
shKBlV5r+s6DiebcTPuLeBtxuML0r+X4j6P7dFR6w0wr4Jc1XfaNq
fTqYHaWN0UyS37Cdp/oL+aIloizG/mNcNGt6K8mYN9DGzeQls0XIk
qCX4Hh6BAoGBAP3ah/+2No9pdiKWygtcNcuyWydl9PmnEs7Nad5/c
ISzbpmo9KSweLW11UiQUqz0ytzWtw9HUJyXtWMbF20/WBckNnUgAz
WruekA7WY59qlnrboQIXtVa6acgfLJaaDBQce6hqLkz+PyepFcRnY
HvDHIgbDULVEEmvlSn1HM85xKwBoZcZ8bc8MXRR4/g5oWygnevf6b
tJtbTXRHeJ7pPXDkm/gHog1n6L3fmJ9AwBhUF0maidJem1S8mqrMD
j8t5gdtAOA9v79PZBvk8Dq5pO5tOgwYJemh1OBl5AoGBAMzQ3fvOe
U9022V8PXzlyuLUMfUfc+Z/HowOogJlPZ3z29+0tzaNf8c0n5I9h8
Y0yxlHhEOS/QAbj2+BiZNhC1LYYdwHkopB9OzvJ5vcVozWLxA5ZZp
A1vSo1791L+gPzpzrTFbLeEJAoGASlaV80H9LWri1N+AZL7xDh+NF
MDnojUKblKWRo2uNjIG0ZM+Q8FF1nwssMBC8/7CU9QIl9xY55TxG9
nUKio30aPPdQQam6dsodc[illegible]6WR6QSuZnfnDfYwcPgoVzIJevDsU8Ow
NxrO7GkCgYBD3wbrTutblxgH[illegible]oDOdP96dX5uh2UmGrWbeMsYQ5k4u
7pOM4YM+m/E3nUat8A0mGLU+QE[illegible]HiVBv2DDCawK2/VO4CtWPZJZpYl
e9lFO87NltY3M0IN70moBVos4u[illegible]162N2zoOcD9k/0Xcqv4hYydA8E
-----END RSA PRIVATE KEY-----
```

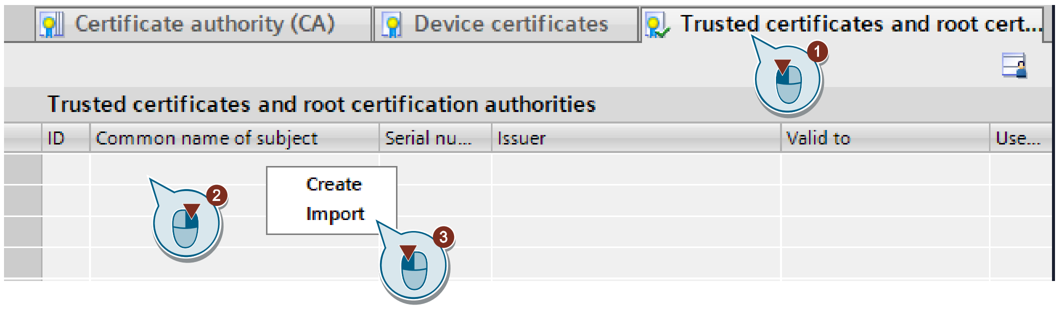
2.5.2 Import CA certificate and device certificate into TIA Portal

To encrypt the MQTT communication between an S7 CPU and AWS IoT with TLS, the amazon root CA certificate ("AmazonRootCA1") and the certificate with the private key (combined to a single file in chapter 2.5.1) must be imported into TIA Portal.

Import AWS CA certificate and device certificate

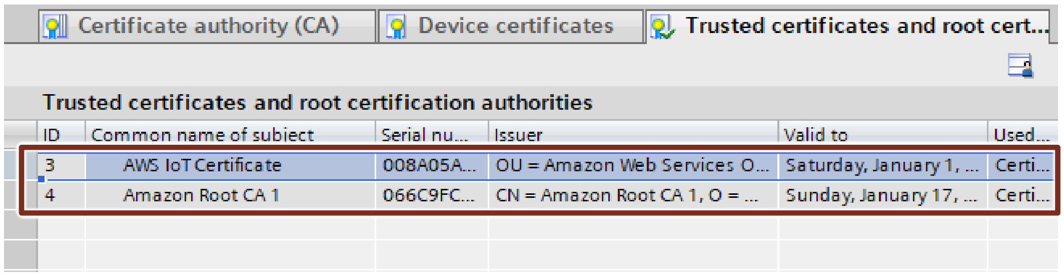
Proceed as follows to import the certificates into TIA Portal:

1. Open the global certificate manager of your TIA Portal project (see chapter 2.1). Select the tab "Trusted certificates and root certification authorities". To open the context menu, right-click in the tab and select "Import".


Certificate authority (CA)
Device certificates
Trusted certificates and root cert...
1
Trusted certificates and root certification authorities
ID
Common name of subject
Serial nu...
Issuer
Valid to
Use...
2
Create
Import
3

2. Navigate to the directory, the certificates are saved. Select the certificate "AmazonRootCA1" and the device certificate you modified in chapter 2.5.1.

3. The certificates are now located in the global certificate manager and have a unique ID.


Certificate authority (CA)
Device certificates
Trusted certificates and root cert...
Trusted certificates and root certification authorities

ID	Common name of subject	Serial nu...	Issuer	Valid to	Used...
3	AWS IoT Certificate	008A05A...	OU = Amazon Web Services O...	Saturday, January 1, ...	Certi...
4	Amazon Root CA 1	066C9FC...	CN = Amazon Root CA 1, O = ...	Sunday, January 17, ...	Certi...

2.5.3 Assign certificates to the S7 CPU

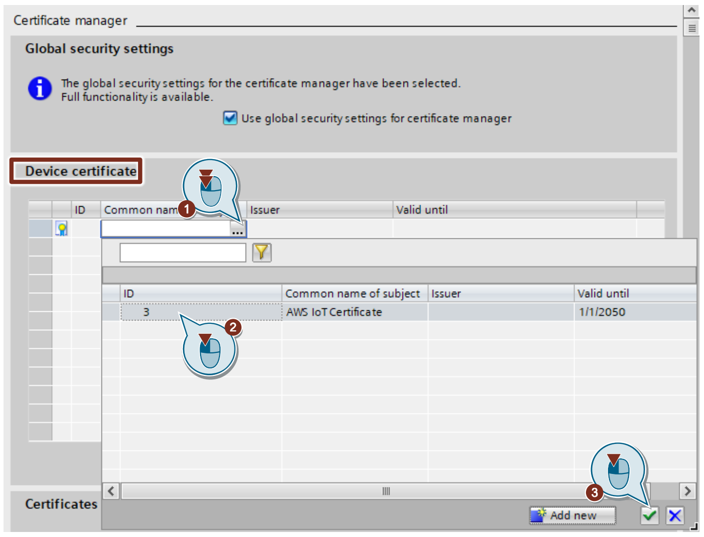
The certificates imported from AWS IoT are now located only in the global certificate manager of TIA Portal and are not automatically assigned to the S7 CPU.

The device certificate and the CA certificate will now be assigned to the local certificate store of the CPU.

Assign device certificate

Proceed as follows to assign the device certificate:

1. Open the local certificate manager of your CPU (see chapter 2.1). Double-click on an empty table row in the "Device certificates" area.
   Clicking on "..." in the new line will open the new certificate selector. Select the AWS IoT certificate and click the green checkmark.



Certificate manager

Global security settings

The global security settings for the certificate manager have been selected.
Full functionality is available.

☑ Use global security settings for certificate manager

Device certificate

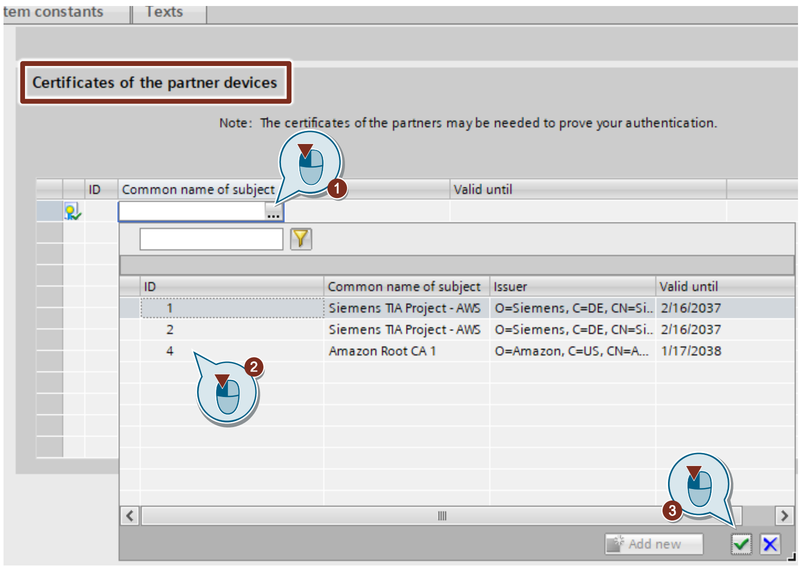
ID	Common name of subject	Issuer	Valid until
3	AWS IoT Certificate		1/1/2050

2. The selected certificate was assigned to the S7 CPU and provided with an ID. The ID is the number of the certificate. You must enter this value in the connection parameters for the tag "tls.ClientCert" (see chapter 2.6).

Assign CA certificate

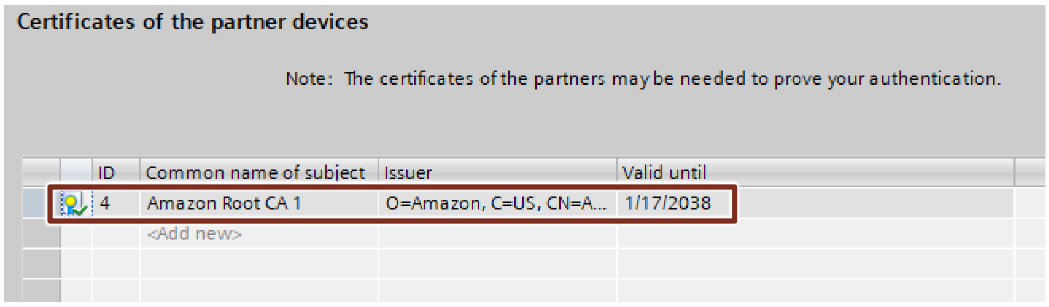
Proceed as follows to assign the CA certificate:

1. Scroll to the area called "Certificates of the partner devices". Double-click on a table row. Clicking on "..." in the new line will open the new certificate selector. In the global certificate manager, select the Amazon root CA certificate (here with ID 4) and click the green checkmark.


tem constants
Texts
Certificates of the partner devices
Note: The certificates of the partners may be needed to prove your authentication.
1
2
3
Add new

ID	Common name of subject	Issuer	Valid until
1	Siemens TIA Project - AWS	O=Siemens, C=DE, CN=Si...	2/16/2037
2	Siemens TIA Project - AWS	O=Siemens, C=DE, CN=Si...	2/16/2037
4	Amazon Root CA 1	O=Amazon, C=US, CN=A...	1/17/2038

2. The selected certificate was assigned to the S7 CPU and provided with an ID. The ID is the number of the certificate. You will enter this value in the connection parameters for the "tls.brokerCert" tag (see chapter 2.6).



Certificates of the partner devices

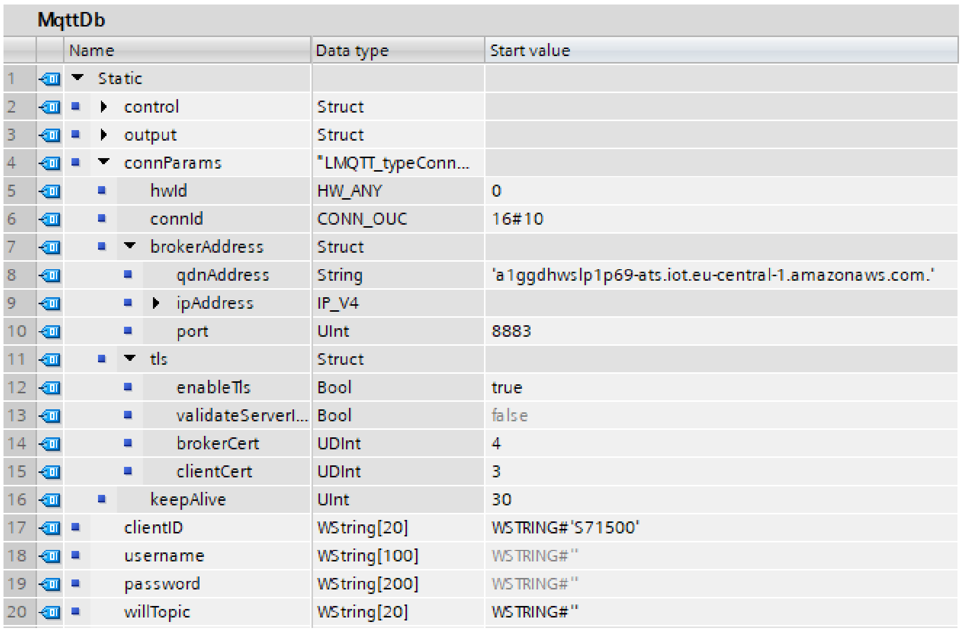
Note: The certificates of the partners may be needed to prove your authentication.

ID	Common name of subject	Issuer	Valid until
4	Amazon Root CA 1	O=Amazon, C=US, CN=A...	1/17/2038
	<Add new>		

2.6 Parameter assignment

Before you can test the application example, you must adjust the parameters for the secure TCP connection and MQTT to suit your use case. The parameters are located in the global data block "MqttDb".

Set the connection parameters as follows:



MqttDb

	Name	Data type	Start value
1	Static		
2	control	Struct	
3	output	Struct	
4	connParams	"LMQTT_typeConn...	
5	hwId	HW_ANY	0
6	connId	CONN_OUC	16#10
7	brokerAddress	Struct	
8	qdnAddress	String	'a1ggdhwslp1p69-ats.iot.eu-central-1.amazonaws.com.'
9	ipAddress	IP_V4	
10	port	UInt	8883
11	tls	Struct	
12	enableTls	Bool	true
13	validateServerI...	Bool	false
14	brokerCert	UDInt	4
15	clientCert	UDInt	3
16	keepAlive	UInt	30
17	clientID	WString[20]	WSTRING#'S71500'
18	username	WString[100]	WSTRING#''
19	password	WString[200]	WSTRING#''
20	willTopic	WString[20]	WSTRING#''

Hardware ID

Set the "hwId" tag to "0" for automatic selection.

Connection ID

The "connID" tag can be given any value. The value must be unique and cannot be used by a different application at the same time.

AWS IoT Core URL

For the "qdnAddress" tag, enter the device data endpoint from chapter 2.4.4. The endpoint name must end with a ".".

Port

For the "port" tag, enter the value "8883" for a secured MQTT connection.

Use of TLS

A valid, imported certificate is a prerequisite for secure communication.

- Set the "enableTls" tag in the "tls" structure to "TRUE".
- Set the "brokerCert" tag in the "tls" structure to the ID of the Amazon CA certificate. It was added to the local certificate manager in chapter 2.5.3.
- Set the "clientCert" tag in the "tls" structure to the ID of the device certificate. It was added to the local certificate manager in chapter 2.5.3.

MQTT client ID

The "clientID" tag is freely selectable.

MQTT user name

The "username" tag is not used.

MQTT password

The tag "password" is not used.

2.7 Operation

Requirements

The following points are prerequisites before testing the application example:

- The project has been downloaded to the CPU.
- The clock time in the CPU is accurate.
- The SIMATIC station is connected to the internet.
- All parameters have been set.

Note The application example uses blocks from the "LMQTT" library. This library is part of the libraries for communication. You can find the library and a detailed block description in Siemens Industry Online Support (see \5\ in chapter 4.3).

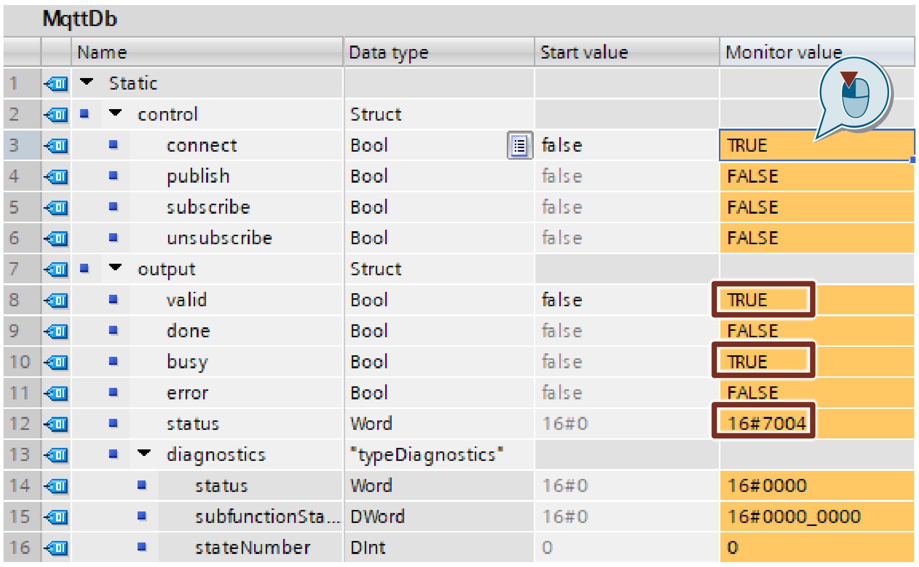
Test the connection

If the requirements are met, the application example can be tested.

Initiate MQTT communication between the CPU and AWS IoT. To do this, trigger the "connect" input in the "MqttDb" data block.

Note As long as the input "connect" is set to "TRUE", the connection will be maintained. If the "connect" input is reset to "FALSE", the connection will be terminated.

If the connection is established successfully, the output tag "status" will be set to the value "16#7004" and the "valid" and "busy" bits will be set to "TRUE".



MqttDb

	Name	Data type	Start value	Monitor value
1	Static			
2	control	Struct		
3	connect	Bool	false	TRUE
4	publish	Bool	false	FALSE
5	subscribe	Bool	false	FALSE
6	unsubscribe	Bool	false	FALSE
7	output	Struct		
8	valid	Bool	false	TRUE
9	done	Bool	false	FALSE
10	busy	Bool	false	TRUE
11	error	Bool	false	FALSE
12	status	Word	16#0	16#7004
13	diagnostics	"typeDiagnostics"		
14	status	Word	16#0	16#0000
15	subfunctionSta...	DWord	16#0	16#0000_0000
16	stateNumber	DInt	0	0

If an error occurs, you can find relevant information in chapter 2.8.

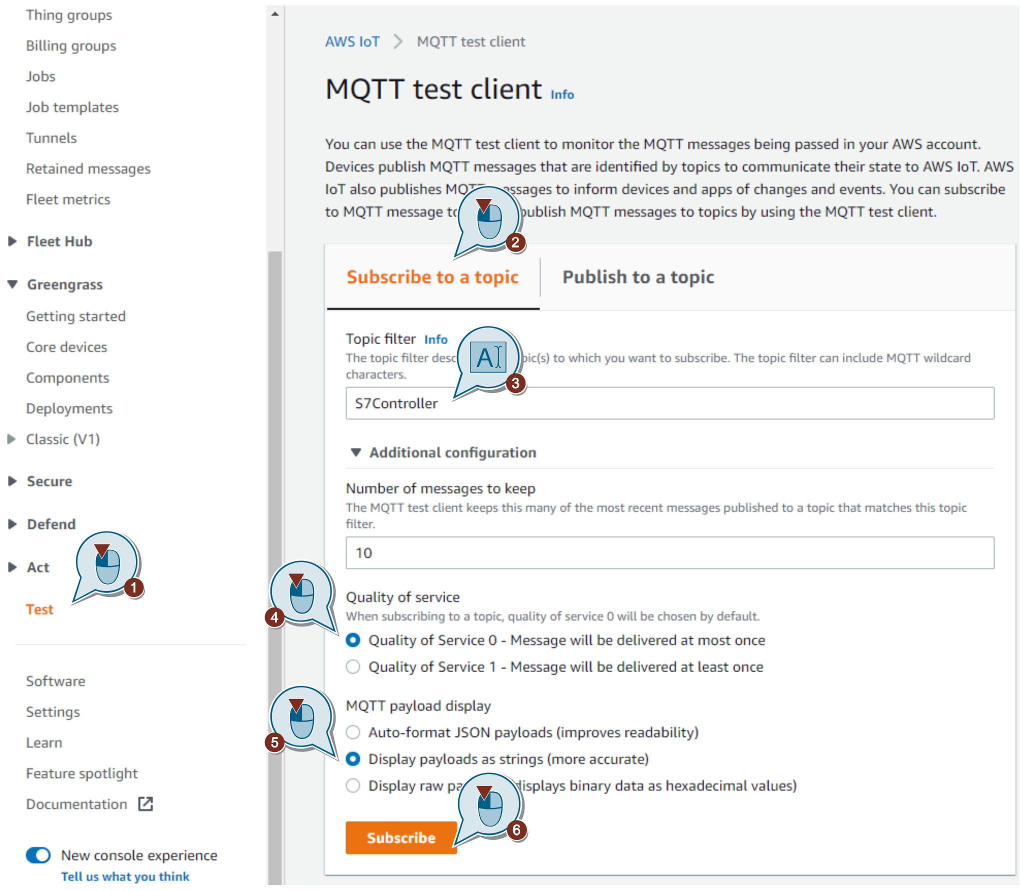
Send MQTT messages to AWS IoT

If the connection is established, you can send messages to AWS IoT with an arbitrary topic name (see \4\ in chapter 4.3).

1. In the lefthand pane of the Amazon Cloud, navigate to "AWS IoT > Test > MQTT test client".

2. Select the "Subscribe to a topic" tab.
   - Enter the topic name in the "Topic filter" field.
   - In the "Additional configuration" menu, select "Quality of Service 0".
   - Set "MQTT payload display" to "Display payloads as strings (more accurate)".

   Click the "Subscribe" button.



MQTT test client Info

You can use the MQTT test client to monitor the MQTT messages being passed in your AWS account. Devices publish MQTT messages that are identified by topics to communicate their state to AWS IoT. AWS IoT also publishes MQ…ssages to inform devices and apps of changes and events. You can subscribe to MQTT message t… publish MQTT messages to topics by using the MQTT test client.

Topic filter Info
The topic filter desc… pic(s) to which you want to subscribe. The topic filter can include MQTT wildcard characters.
S7Controller

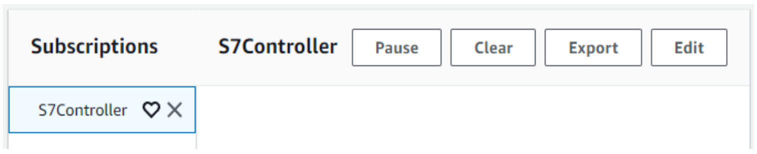
Additional configuration

Number of messages to keep
The MQTT test client keeps this many of the most recent messages published to a topic that matches this topic filter.
10

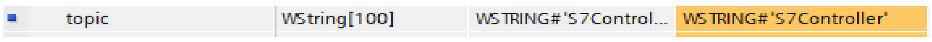
Quality of service
When subscribing to a topic, quality of service 0 will be chosen by default.
[✓ Quality of Service 0 - Message will be delivered at most once]
Quality of Service 1 - Message will be delivered at least once

MQTT payload display
Auto-format JSON payloads (improves readability)
[✓ Display payloads as strings (more accurate)]
Display raw pa… displays binary data as hexadecimal values)

The MQTT test client in AWS is ready to receive messages with the subscribed topic.

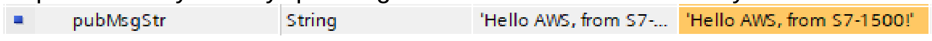


3. Switch to TIA Portal and, in the "topic" tag in the "MqttDb" data block, enter the topic name that you subscribed to in Step 1.



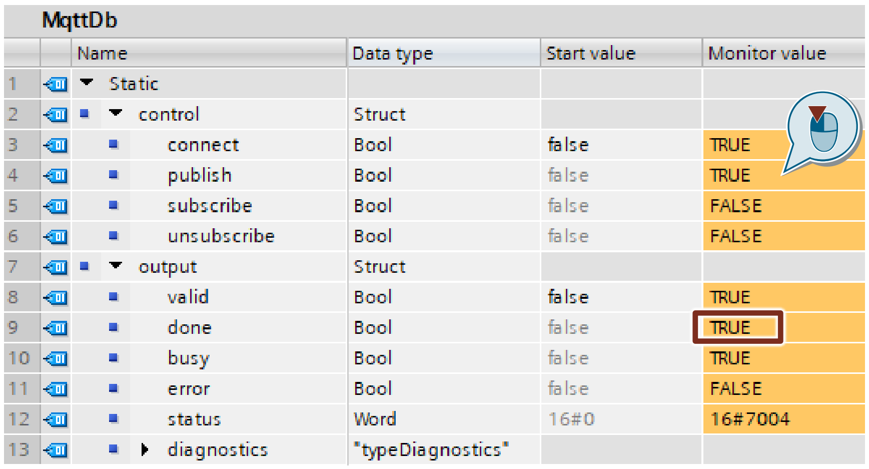
topic	WString[100]	WSTRING#'S7Control...	WSTRING#'S7Controller'

4. Populate the byte array "pubMsgStr" with the information that you wish to send.



pubMsgStr	String	'Hello AWS, from S7-...	'Hello AWS, from S7-1500!'

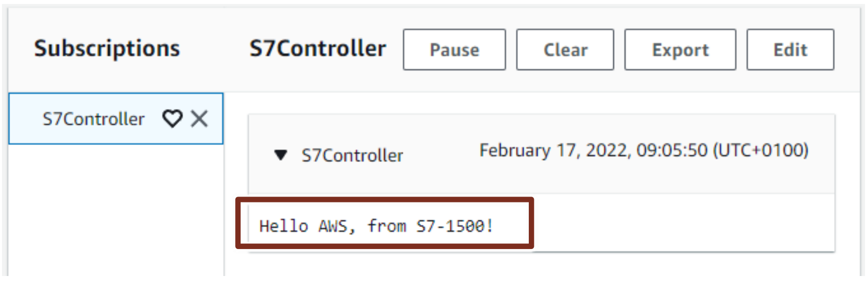
5. The message will be sent when you set the "publish" tag to "TRUE". If the transmission was successful, the output bit "done" will be set to "TRUE".



MqttDb

	Name	Data type	Start value	Monitor value
1	▼ Static			
2	▼ control	Struct		
3	connect	Bool	false	TRUE
4	publish	Bool	false	TRUE
5	subscribe	Bool	false	FALSE
6	unsubscribe	Bool	false	FALSE
7	▼ output	Struct		
8	valid	Bool	false	TRUE
9	done	Bool	false	TRUE
10	busy	Bool	false	TRUE
11	error	Bool	false	FALSE
12	status	Word	16#0	16#7004
13	▶ diagnostics	"typeDiagnostics"		

6. Go to AWS IoT in the "Subscriptions" view to see the sent message.



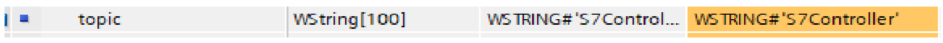
February 17, 2022, 09:05:50 (UTC+0100)

```
Hello AWS, from S7-1500!
```

Receive MQTT messages from AWS

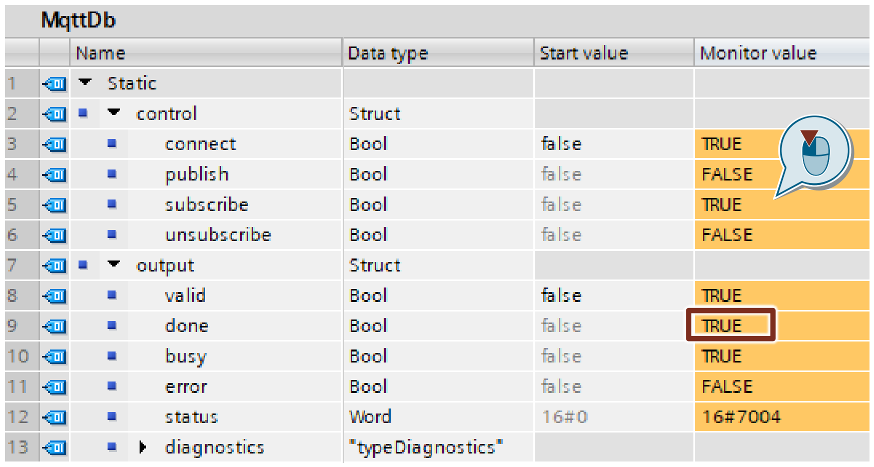
To receive messages from an IoT Hub, subscribe to a topic of your choice.

1. In the "topic" tag in the "MqttDb" data block, enter a topic name (it can be the same name as the one in chapter 2.5.2).



topic	WString[100]	WSTRING#'S7Control...	WSTRING#'S7Controller'

2. To receive messages, set the "subscribe" tag to "TRUE". When a message has been received, the output bit "done" will be set to "TRUE".



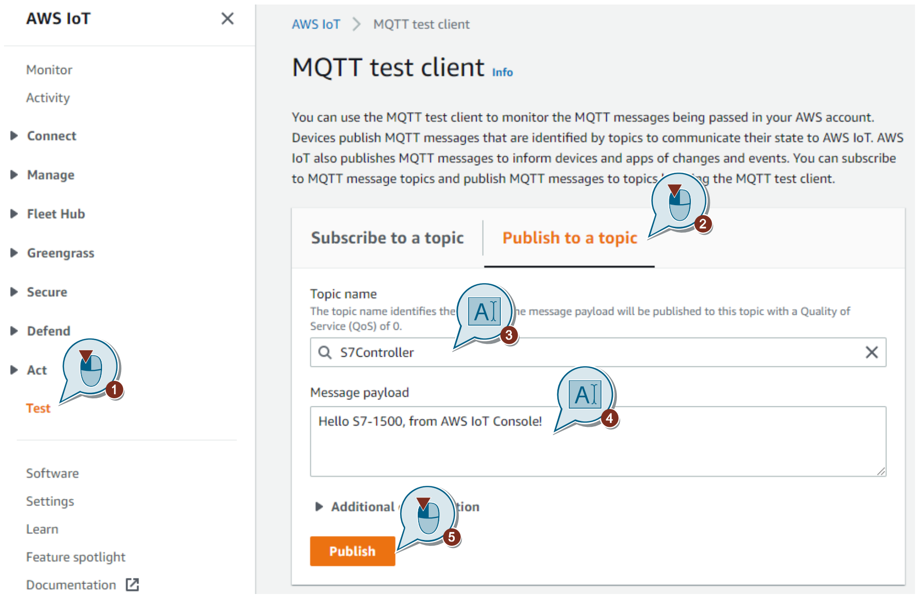
MqttDb

	Name	Data type	Start value	Monitor value
1	▼ Static			
2	▼ control	Struct		
3	connect	Bool	false	TRUE
4	publish	Bool	false	FALSE
5	subscribe	Bool	false	TRUE
6	unsubscribe	Bool	false	FALSE
7	▼ output	Struct		
8	valid	Bool	false	TRUE
9	done	Bool	false	TRUE
10	busy	Bool	false	TRUE
11	error	Bool	false	FALSE
12	status	Word	16#0	16#7004
13	▶ diagnostics	"typeDiagnostics"		

3. In the left-hand pane of the Amazon Cloud, navigate to "AWS IoT > Test > MQTT test client". Select the "Publish to a topic" tab.

   In the "Topic name" field, enter the topic name. In "Message payload", enter the message you wish to send.

   Send the message with "Publish".

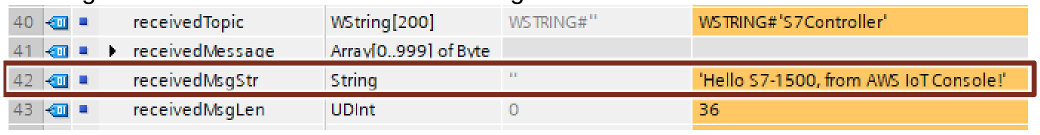


You can use the MQTT test client to monitor the MQTT messages being passed in your AWS account. Devices publish MQTT messages that are identified by topics to communicate their state to AWS IoT. AWS IoT also publishes MQTT messages to inform devices and apps of changes and events. You can subscribe to MQTT message topics and publish MQTT messages to topics using the MQTT test client.

Topic name
The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.
S7Controller

Message payload
Hello S7-1500, from AWS IoT Console!

4. Go to TIA Portal in the "MqttDb" data block and here to "receivedMsgStr" to see the message received from AWS IoT as a string.



40	receivedTopic	WString[200]	WSTRING#''	WSTRING#'S7Controller'
41	receivedMessage	Array[0..999] of Byte		
42	receivedMsgStr	String	''	'Hello S7-1500, from AWS IoT Console!'
43	receivedMsgLen	UDInt	0	36

2.8 Error handling

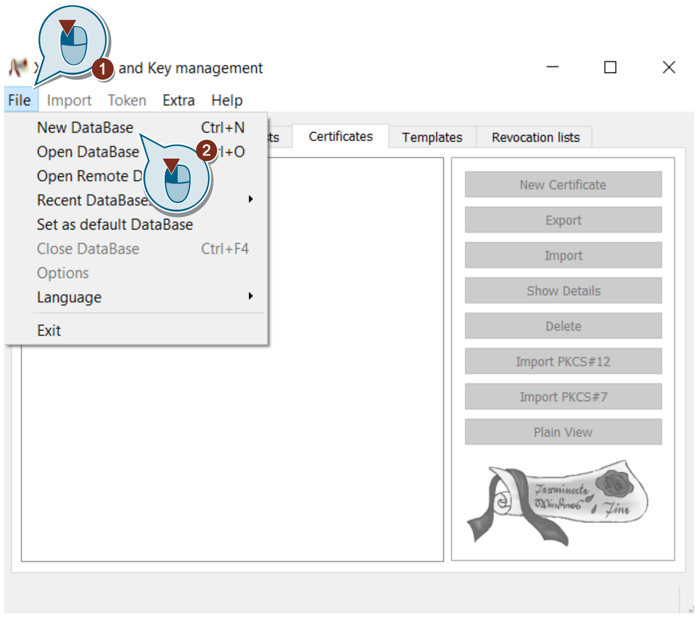
For a description of each error code, refer to the documentation on the Libraries for Communication (see \5\ in chapter 4.3).

3 Additional information

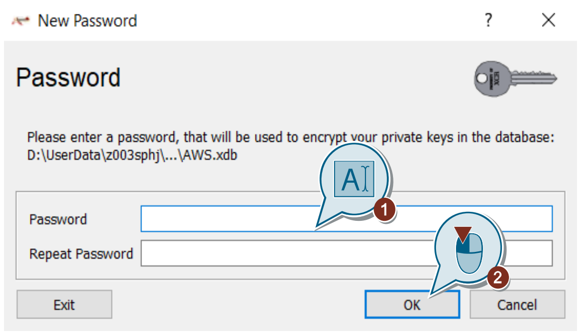
The "X-Certificate and Key management" software is used to create and manage X509 certificates. Proceed as follows to combine a certificate with its private key and get an encrypted file:

1. Download the "XCA" software (see \4\ in chapter 4.3) and install it.

2. Open XCA and create a new database.

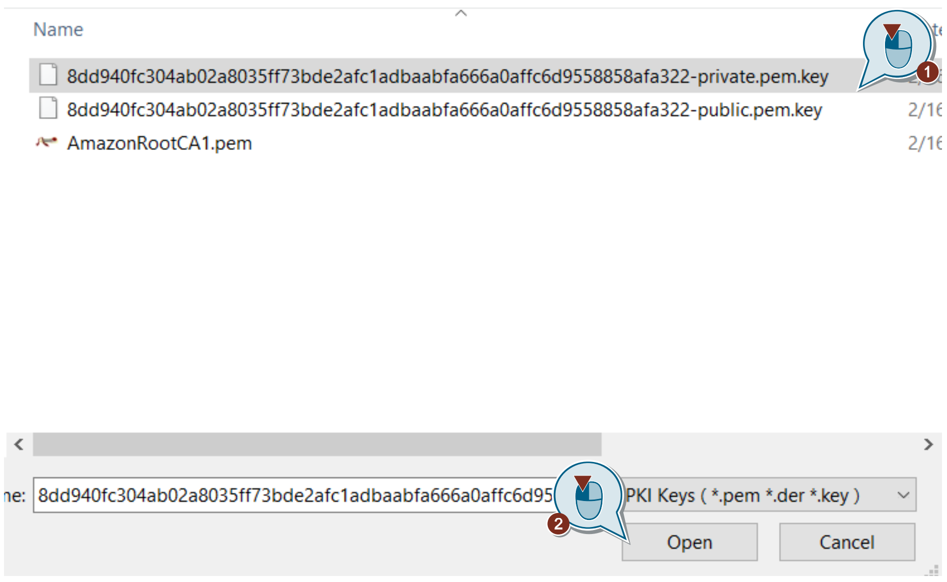


3. Enter a name for the database and enter a password that will be used to encrypt the private key. Click "OK".



Please enter a password, that will be used to encrypt your private keys in the database:
D:\UserData\z003sphj\...\AWS.xdb

4. Select "Import > Key" and import the private key for the certificate.



Name

8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322-private.pem.key

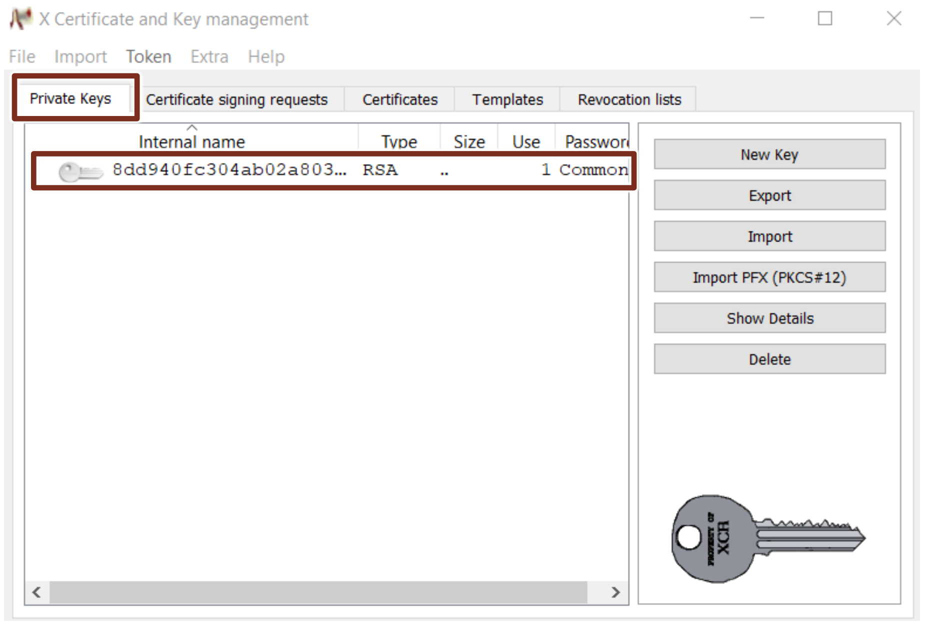
8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322-public.pem.key 2/16

AmazonRootCA1.pem 2/16

ne: 8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d95

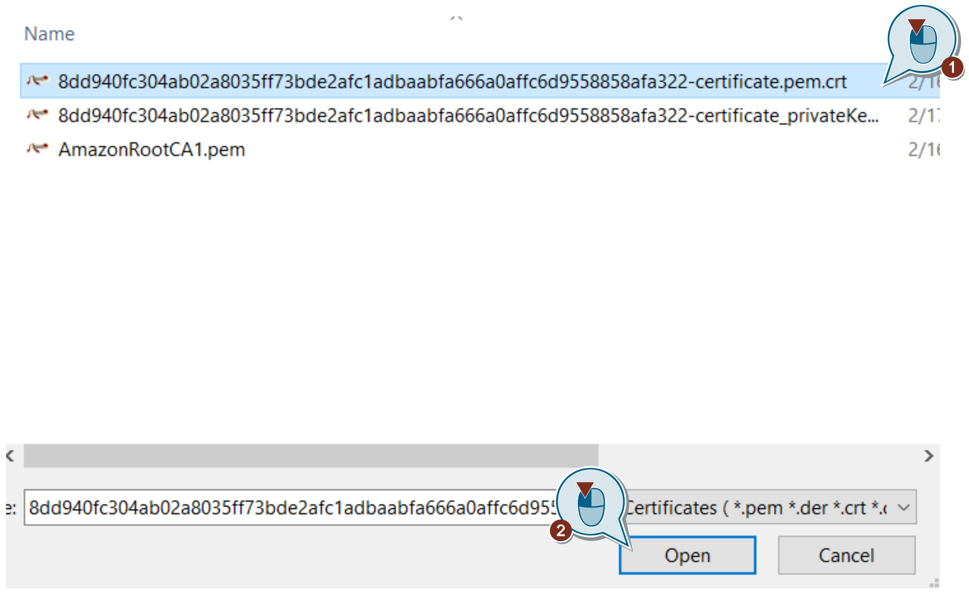
PKI Keys (*.pem *.der *.key)

5. The imported key will appear in the "Private Keys" tab.



Internal name	Type	Size	Use	Passwor
8dd940fc304ab02a803…	RSA	..	1	Common

6. Select "Import > Certificates" and import the device certificate from AWS.



Name

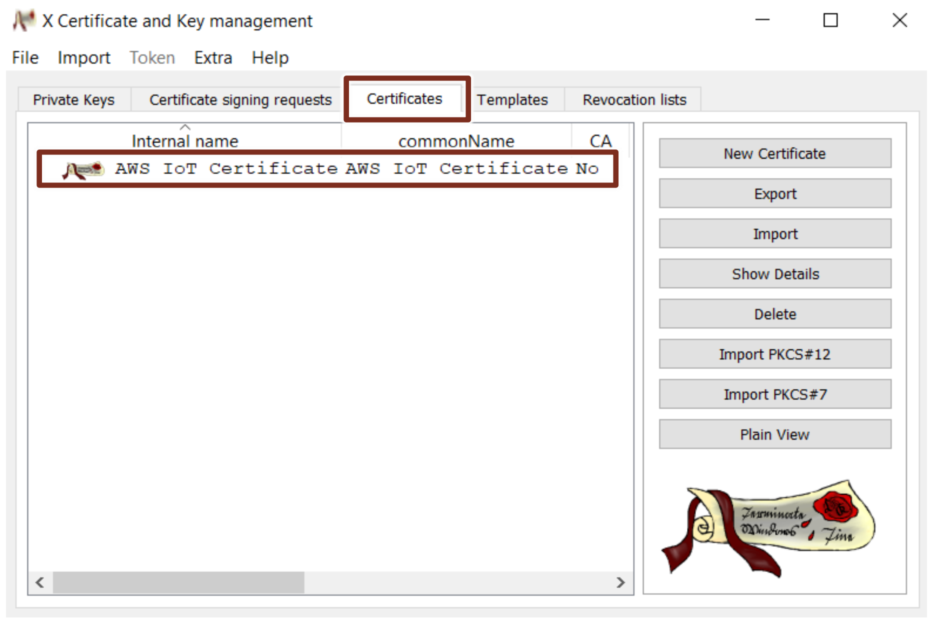
8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322-certificate.pem.crt

8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d9558858afa322-certificate_privateKe...

AmazonRootCA1.pem

1
2
8dd940fc304ab02a8035ff73bde2afc1adbaabfa666a0affc6d955
Certificates (*.pem *.der *.crt *.
Open
Cancel

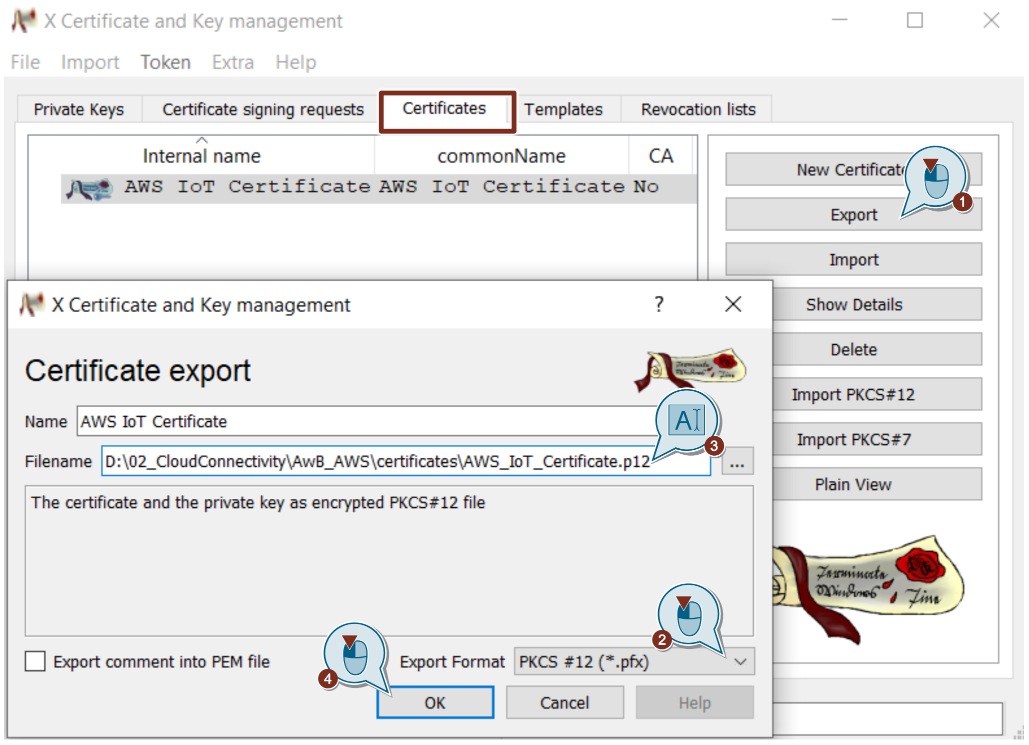
7. The imported certificate will appear in the "Certificates" tab.



X Certificate and Key management

Internal name	commonName	CA
AWS IoT Certificate	AWS IoT Certificate	No

8. Click the "Export" button in the right-hand pane and select the "PKCS #12 (*.pfx)" export format. Make sure that the file ending is "*.p12" and not "*.pfx".



X Certificate and Key management

Certificate export

AWS IoT Certificate AWS IoT Certificate No

Name: AWS IoT Certificate

Filename: D:\02_CloudConnectivity\AwB_AWS\certificates\AWS_IoT_Certificate.p12

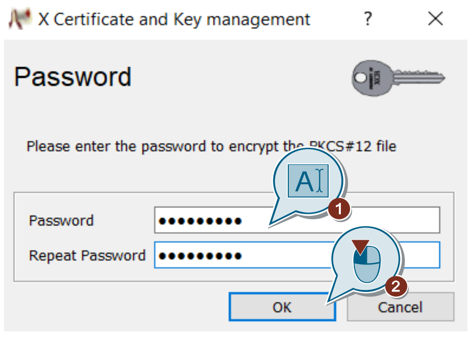
The certificate and the private key as encrypted PKCS#12 file

Export comment into PEM file

Export Format: PKCS #12 (*.pfx)

Note All systems involved must support the PKCS#12 standard.

9. Enter a password to encrypt the PKCS#12 file.



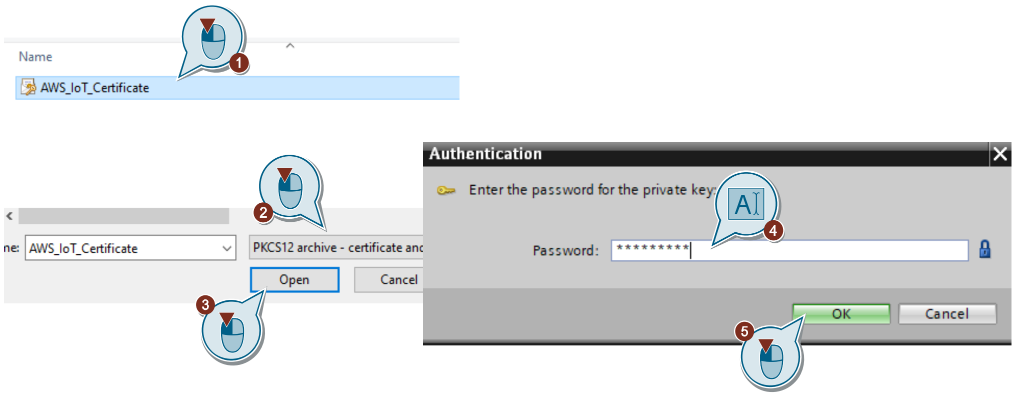
Password

Please enter the password to encrypt the PKCS#12 file

Password: •••••••••

Repeat Password: •••••••••

10. Switch to TIA Portal and import the PKCS12 archive under "Security settings > Security features > Certificate manager > Trusted certificates and root certification authorities". The password you used to encrypt the file will be required again.



11. Follow the instructions in chapter 2.3.2 for how to proceed next.

4 Appendix

4.1 Service and support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:

support.industry.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers
– ranging from basic support to individual support contracts. Please send queries to Technical Support via Web form:

siemens.com/SupportRequest

SITRAIN – Digital Industry Academy

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that's tailored to the customer's specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:

siemens.com/sitrain

Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog web page:

support.industry.siemens.com/cs/sc

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for iOS and Android:

support.industry.siemens.com/cs/ww/en/sc/2067

4.2 Industry Mall

The Siemens Industry Mall is the platform on which the entire siemens Industry product portfolio is accessible. From the selection of products to the order and the delivery tracking, the Industry Mall enables the complete purchasing processing – directly and independently of time and location:
mall.industry.siemens.com

4.3 Links and literature

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to the article page of the application example https://support.industry.siemens.com/cs/ww/en/view/109772284
\3\	Security in AWS IoT https://docs.aws.amazon.com/iot/latest/developerguide/security.html
\4\	XCA tool download: https://hohnstaedt.de/xca/
\5\	Libraries for Communication https://support.industry.siemens.com/cs/de/en/view/109780503
\6\	Application example: Use the SIMATIC controller as an MQTT client https://support.industry.siemens.com/cs/de/en/view/109748872

4.4 Change documentation

Version	Date	Modifications
V1.0	05/2022	First version