

# AWS IoT

## User Manual

*Your industrial control solutions source*

**Table of Contents**

**Overview of AWS IoT.....3**

**Hosting MQTT Server.....3**

**EBPro Settings.....9**

**Thing and Shadow.....12**

**EBPro.....15**

## Overview of AWS IoT

AWS (Amazon Web Service) is a cloud platform now widely used on the market, and AWS IoT (Internet of Things) supports MQTT protocol. Observing the market trend, from EBPro V6.00.01, Maple Systems has adopted AWS IoT service and integrated it with the MQTT feature released earlier. Apart from using AWS IoT as a broker in the publish-subscribe mode, users can also create Thing and Shadow offered by AWS IoT to make the most of MQTT.

This manual covers hosting a MQTT server, configuring EBPro, and establishing an IoT.

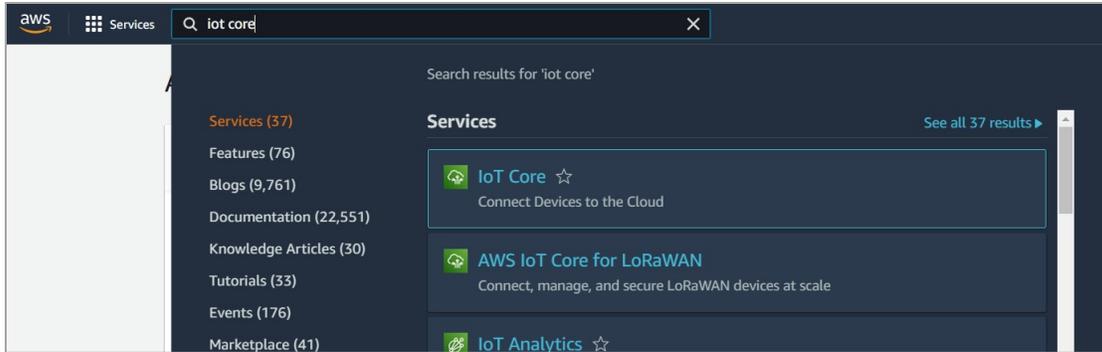
## Hosting MQTT Server

AWS is a cloud platform; therefore, the settings are all configured on the web. Please sign up on the Amazon website before hosting an MQTT server.

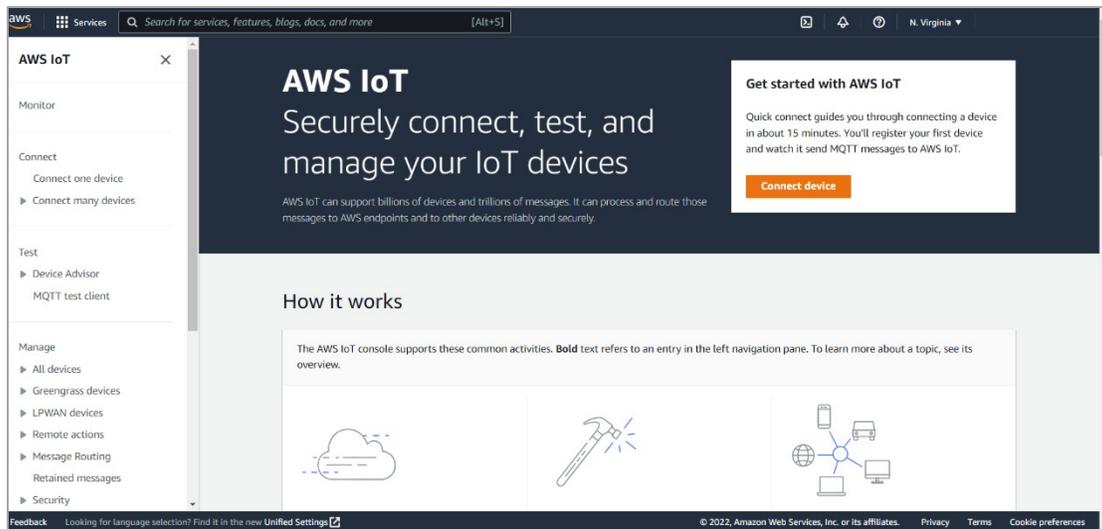
1. Visit Amazon Web Service at <https://aws.amazon.com> and sign up.

The image displays two screenshots of the AWS IAM console sign-in pages. The left screenshot shows the 'Root user sign in' page, which includes the AWS logo, the title 'Root user sign in', and input fields for 'Email' (masked with asterisks) and 'Password'. A blue 'Sign in' button is located below the password field. At the bottom, there are links for 'Sign in to a different account' and 'Forgot your password?'. The right screenshot shows the 'Sign in as IAM user' page, which includes the AWS logo, the title 'Sign in as IAM user', and input fields for 'Account ID (12 digits) or account alias', 'IAM user name', and 'Password'. A checkbox for 'Remember this account' is present below the password field. A blue 'Sign in' button is located at the bottom of the page.

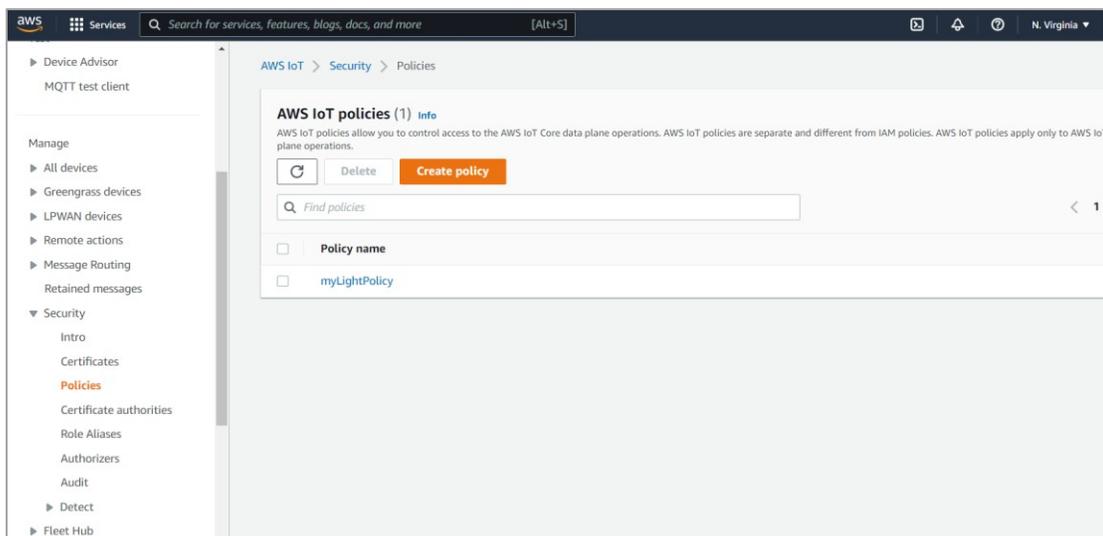
2. After signing in, browse for IoT Core.

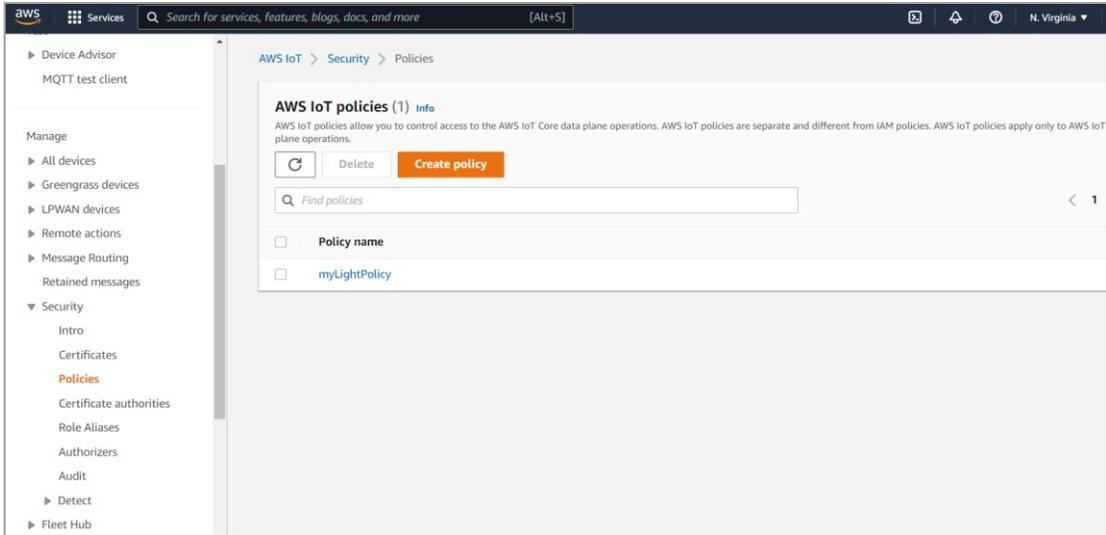


3. On the AWS IoT page, Policy and Certificate can be created.

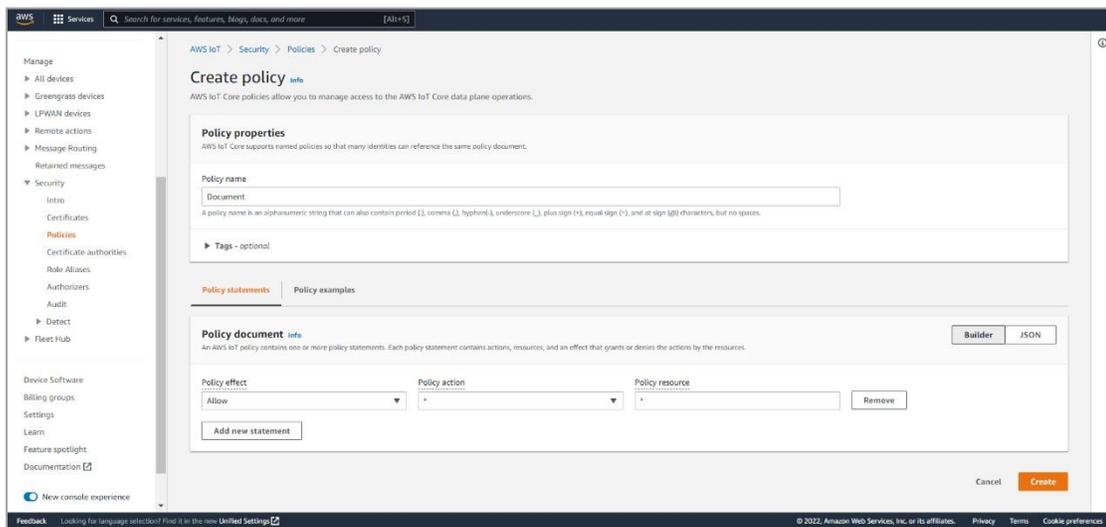


4. Open **Manage** > **Security** > **Policies** and then click **Create policy**.

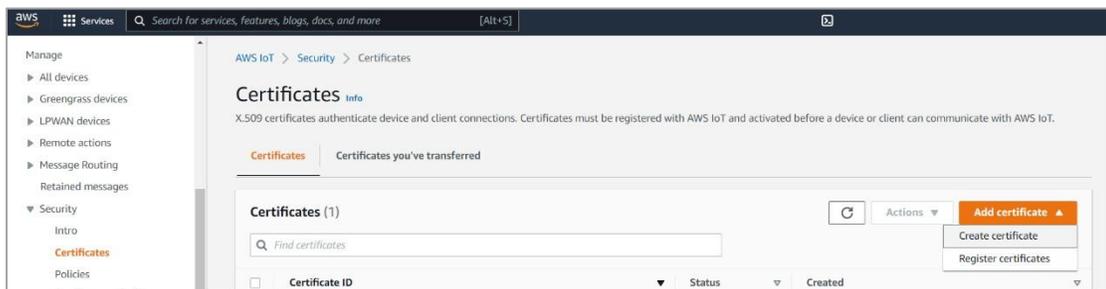




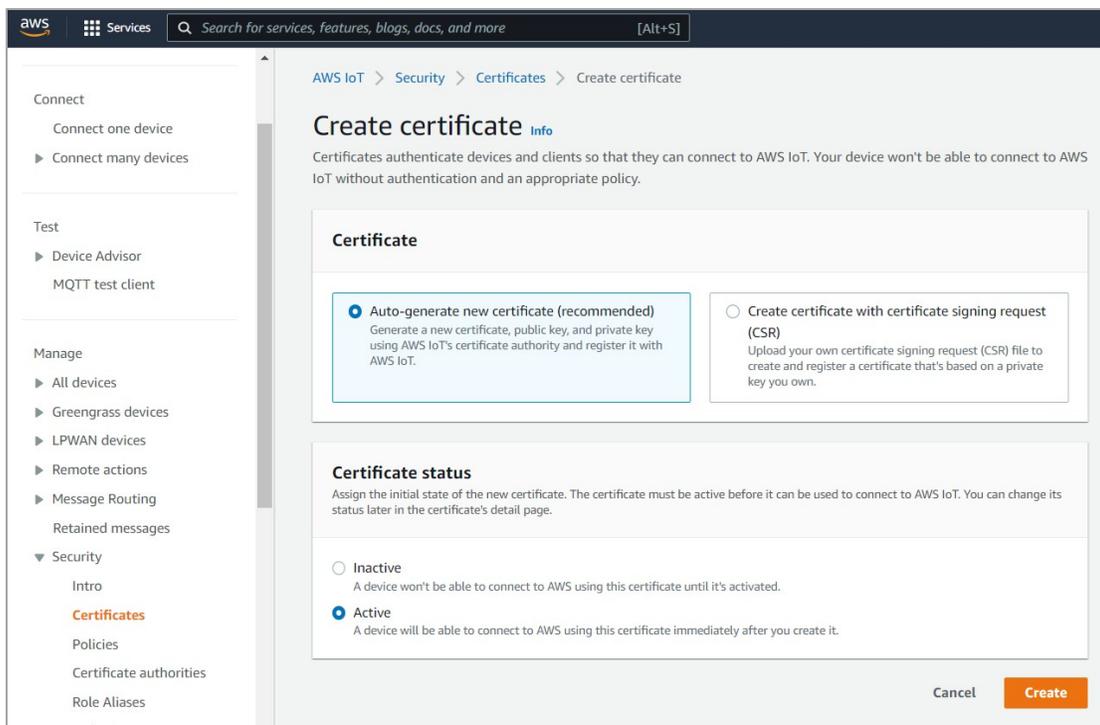
- This page is for defining actions that can be performed by a resource. You may use the settings in the screenshot below or set your own. Click **Create** when finished.



- Click **Security > Certificates** and then click **Add certificate > Create certificate**.



7. Select **Auto-generate new certificate (recommended)**, set Certificate Status to **Active**, and then click **Create**.



8. Download and save these files: **Device Certificate**, **Public Key File**, **Private Key File**, and **RSA 2048 bit key: Amazon Root CA 1**.

### Download certificates and keys ✕

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

Device certificate  Download  
 9c3c9550dfd...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  Download  
 9c3c9550dfdb7324bd36782...d1c4fd-public.pem.key

Private key file  Download  
 9c3c9550dfdb7324bd36782...d1c4fd-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  Download  
 RSA 2048 bit key: Amazon Root CA 1

Amazon trust services endpoint  Download  
 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 from our developer guides.

Continue

9. Click the certificate created previously and click **Attach policies** under Policies. In the window that follows, select the Policy created previously and then click **Attach policies**.

### Attach policies to the certificate ✕

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

Choose AWS IoT policy

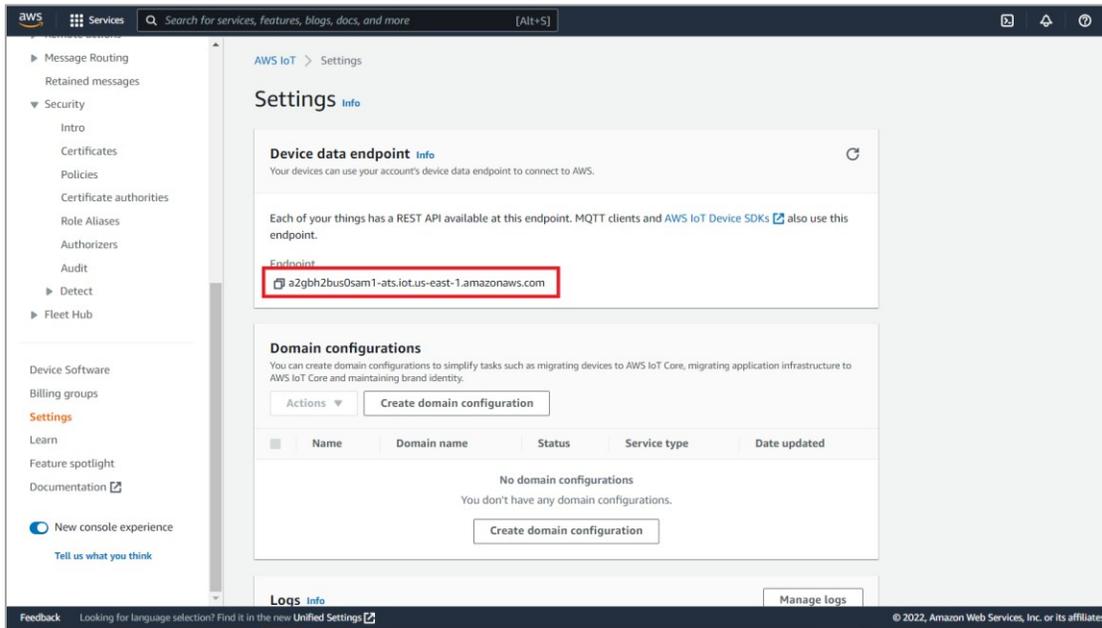
Document ✕

Cancel Attach policies

- Security setting is done successfully when the following box shows:  
Successfully attached the policy Document to certificate



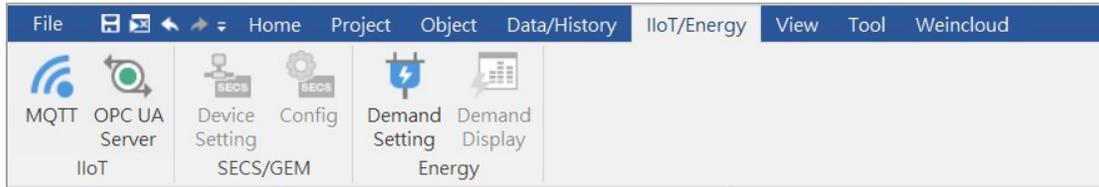
- Click **Settings**. The URL marked in the red frame below is the domain name of the AWS IoT server and will be used when setting MQTT in EBPro; please remember it. AWS is gradually replacing servers using Symantec CA with Amazon Trust Service. As a result, please check whether the domain name contains "-ats", for example: a2xxxxxxx-ats.iot.xxxxxx.amazon.aws.com. The Amazon Root CA 1 certificate created in step 9 works only when the endpoint is in this format.



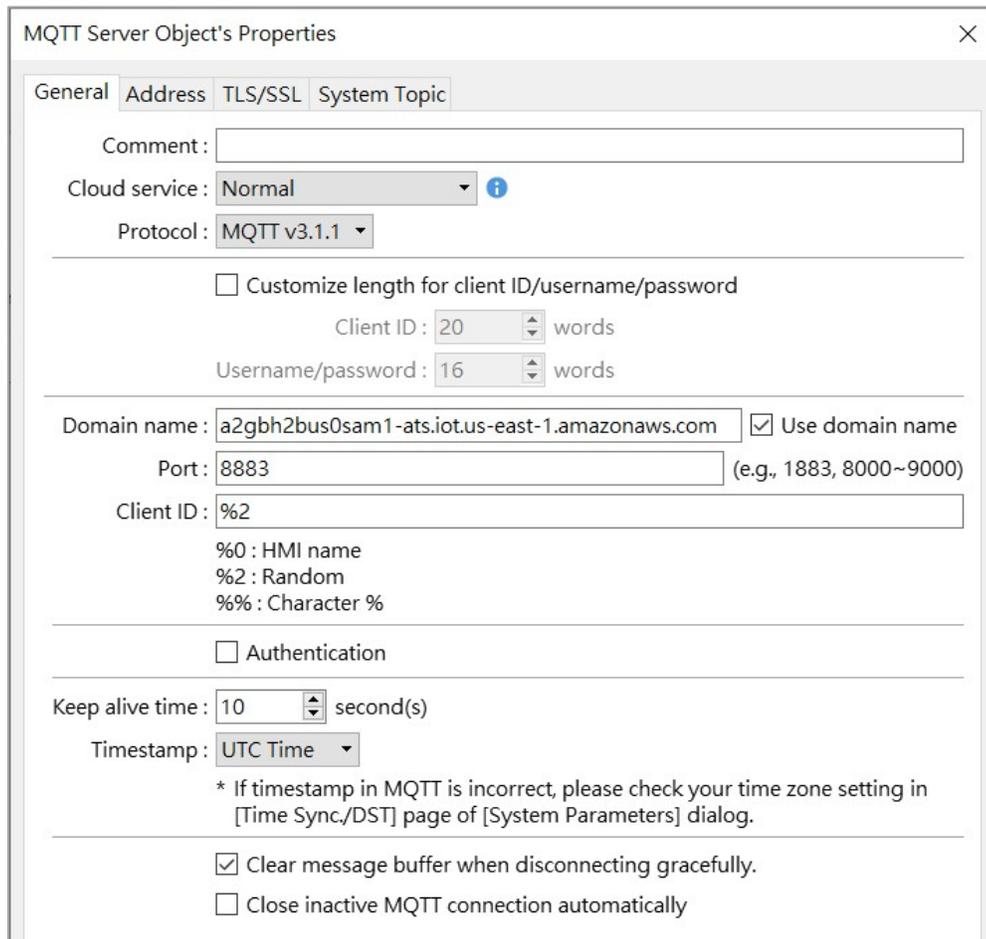
## EBPro Settings

After hosting an MQTT server, launch EBPro.

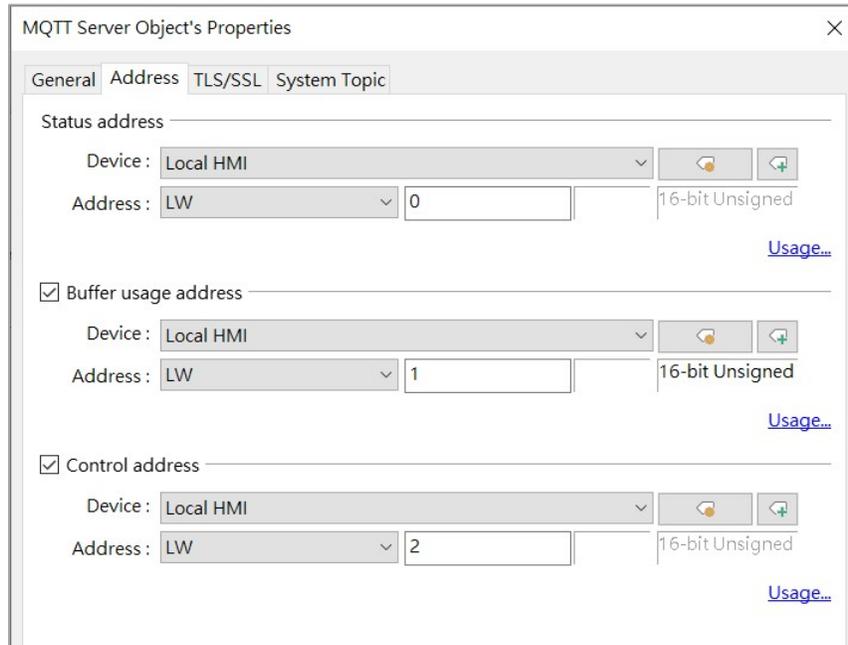
1. Click **IIoT/Energy** > **MQTT** to open the MQTT settings window.



2. In the General tab, select **Normal** as cloud service to use publish-subscribe mode, or select **AWS IoT** to use Thing mode, and the rest will be introduced later. Use the URL obtained in Chapter 2 as domain name and use port 8883.

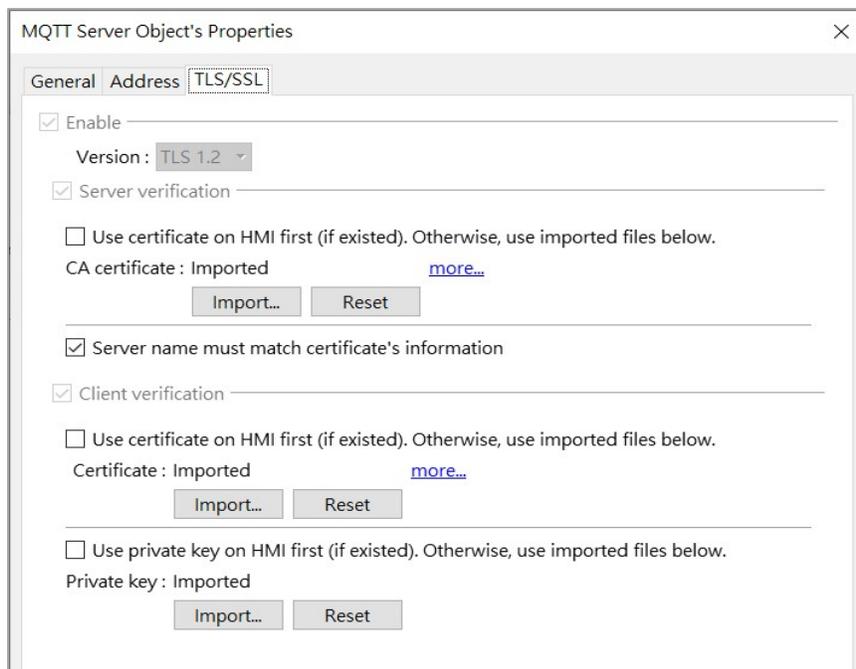


3. Configure addresses in the **Address** tab.

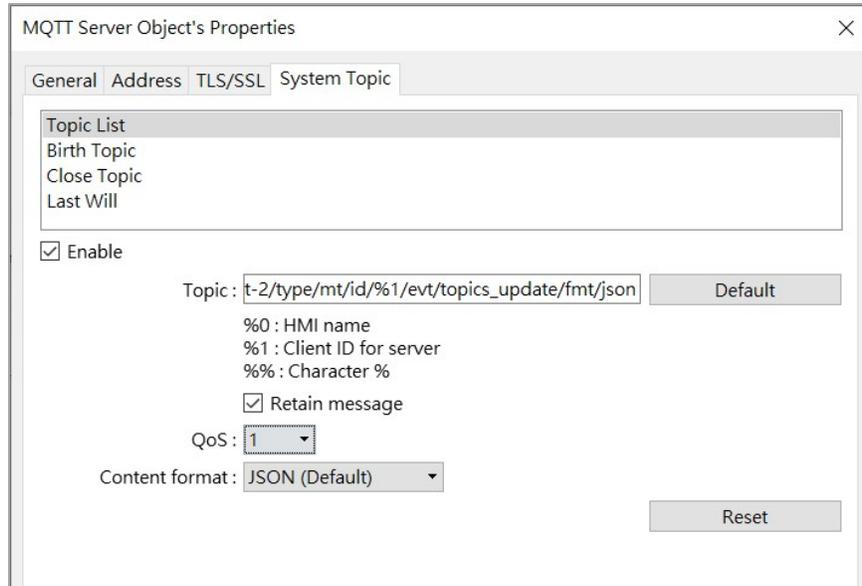


4. In the **TLS/SSL** tab, import the file generated when creating the certificate.

- Server verification, CA certificate: Import a .pem file. (Amazon Root CA 1)
- Client verification, Certificate: Import a .crt file. (certificate.pem.crt)
- Client verification, Private key: Import a .key file (private.pem.key)



- The System topic includes Topic List and Connection State that HMI will automatically send once it connects to server.



- Restrictions of using AWS IoT as MQTT server:
  - Only QoS 0 and QoS 1 are available.
  - Retain message is not supported.
  - The maximum number of layers is 8.
- See EBPro user manual for more information for publish and subscribe settings.

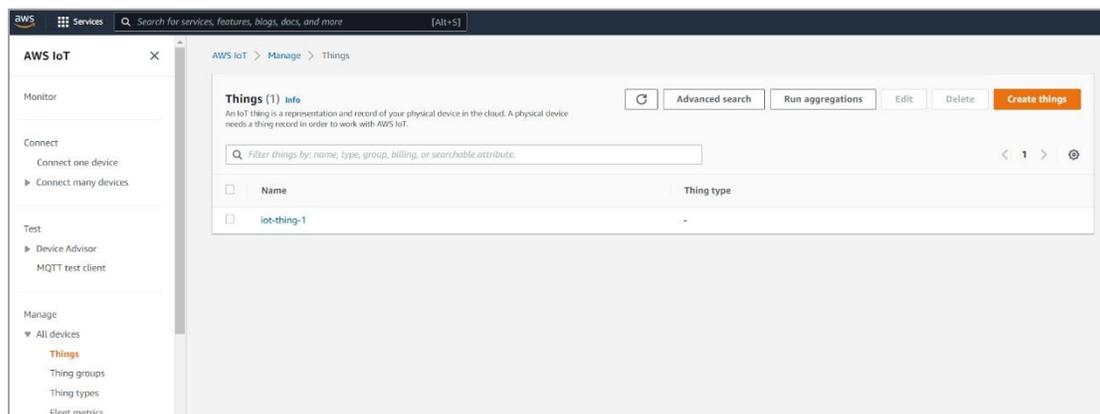
## Thing and Shadow

With AWS IoT, Publisher->Broker->Subscribe is no longer the only path that data is accessed over MQTT. By introducing Thing Shadow service, a Thing (a device, app, etc.) can interact with cloud applications and other devices connected to AWS IoT. A Shadow can be maintained for each Thing connected to AWS IoT. The Shadow can be used to get/set the state of a Thing over MQTT, regardless of whether the Thing is connected to the Internet.

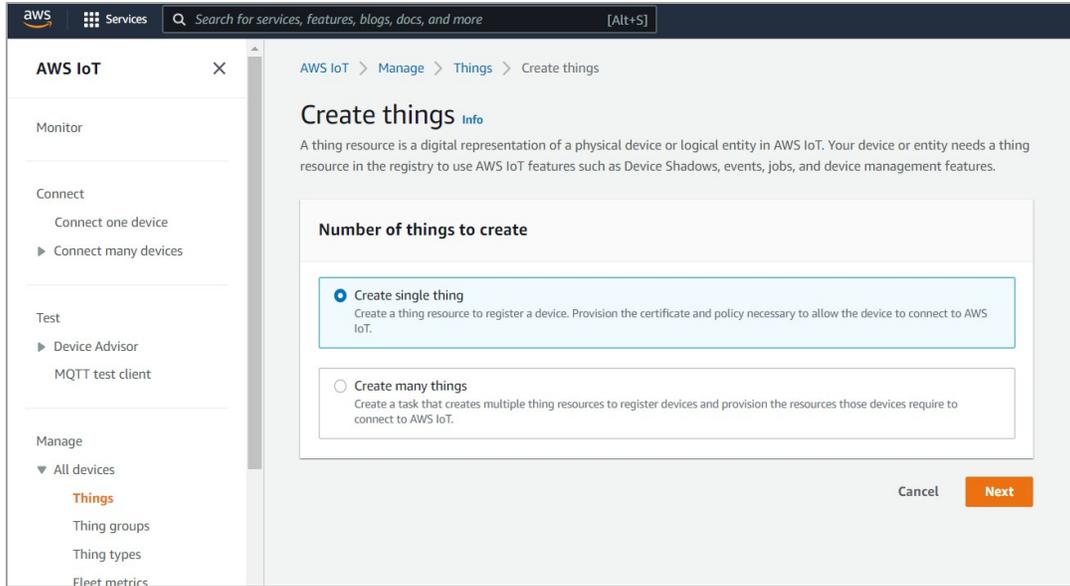
This chapter explains the configuration of AWS IoT and EBPro.

### AWS IoT

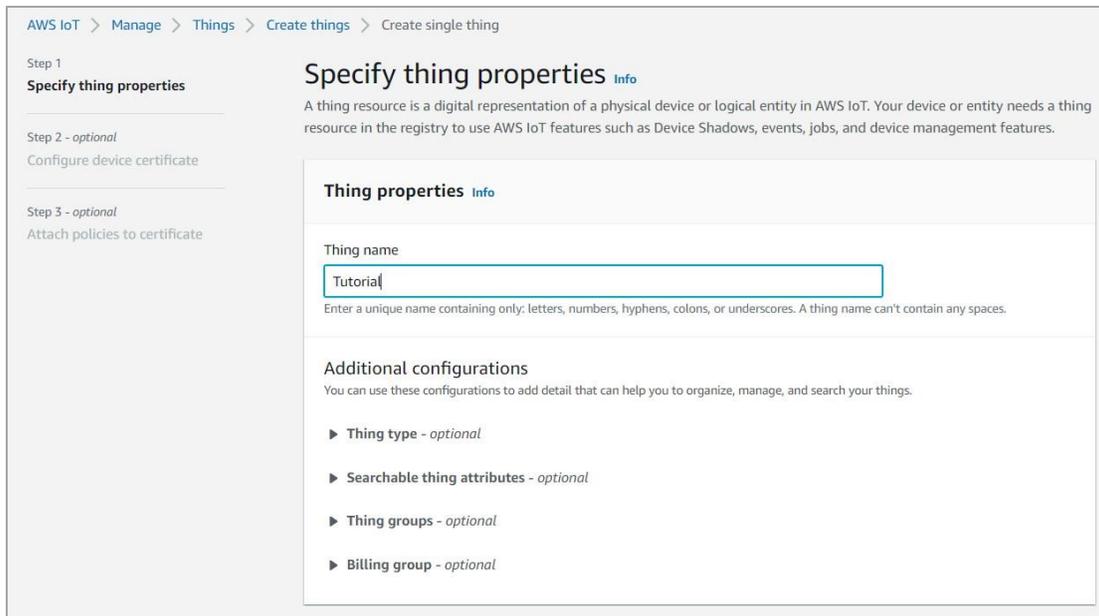
1. Click **Manage > All devices > Things > Create things**.



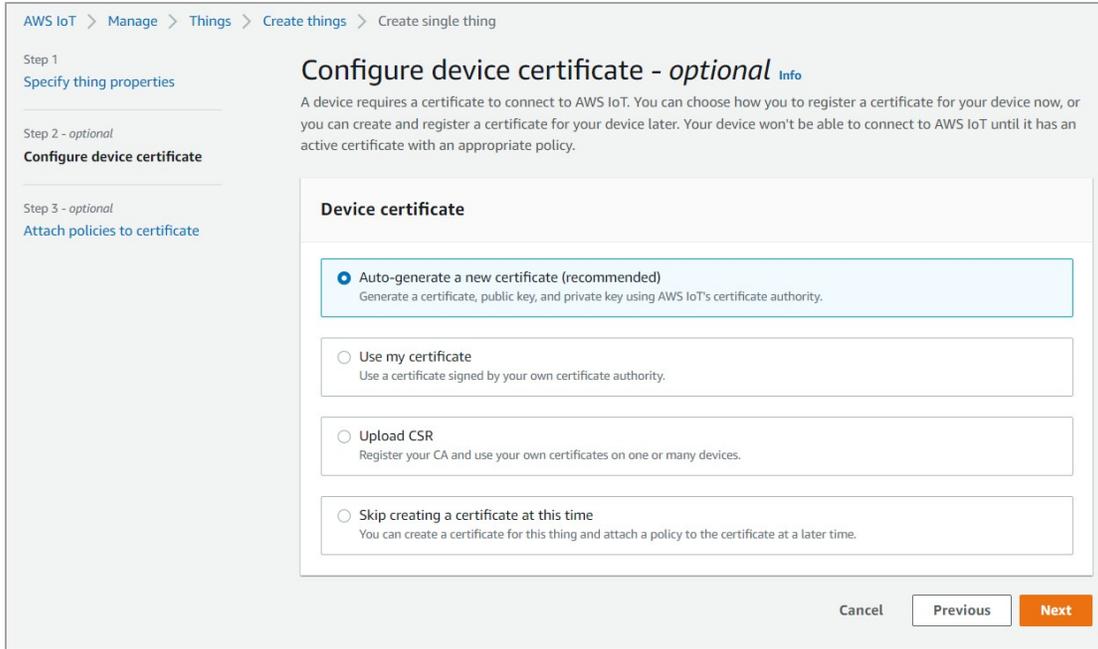
2. Select **Create single thing** and then click **Next**.



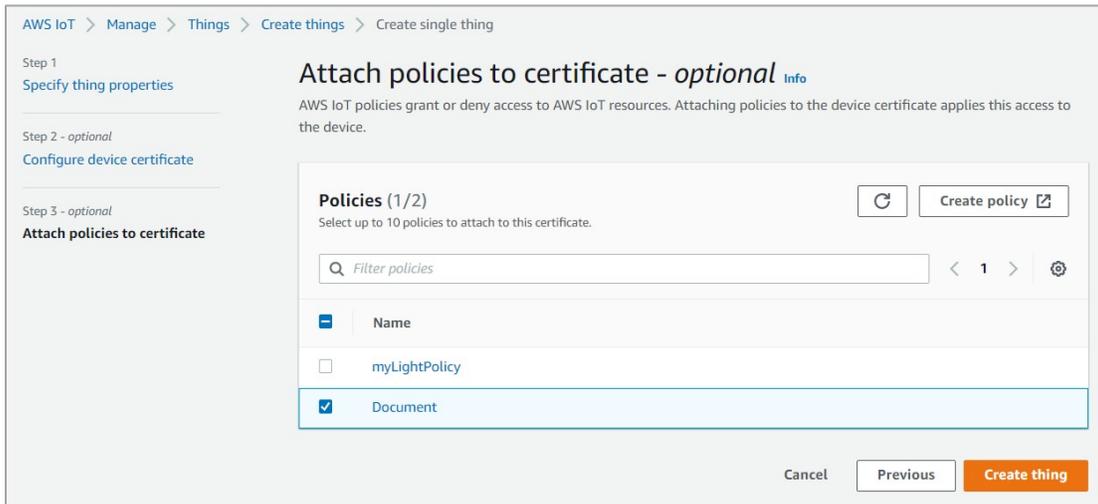
3. Enter the Thing name.



4. Create a certificate.



5. Select the corresponding Policy and then click **Create thing**.



## EBPro

1. Select AWS IoT as cloud service in the MQTT Server settings window and follow the settings in Chapter 3 in this manual.

MQTT Server Object's Properties

General Address TLS/SSL

Comment:

Cloud service: **AWS IoT**

Protocol: **MQTT v3.1.1**

---

Domain name:   Use domain name

Port:  (e.g., 1883, 8000~9000)

Client ID:

%0 : HMI name  
 %2 : Random  
 %% : Character %

---

Keep alive time:  second(s)

Timestamp: **UTC Time**

\* If timestamp in MQTT is incorrect, please check your time zone setting in [Time Sync./DST] page of [System Parameters] dialog.

2. Click New to add a device.

MQTT

Enable

Server

Domain name : a2gbh2bus0sam1-ats.iot.us-east-1.amazonaws.com, Port : 8883

AWS IoT Thing

Thing Name	Description

3. Enter Thing name and set minimal time between messages. Only QoS 0 and QoS 1 are available.

The screenshot shows a dialog box titled "New AWS IoT Thing Object" with a close button (X) in the top right corner. It has two tabs: "General" (selected) and "Address". The "General" tab contains the following fields and options:

- Description: [Empty text box]
- Thing name: [default]
- Min. time between messages: 0 [ms] (with a spinner control)
- QoS: 1 (dropdown menu)
- Content format: JSON (dropdown menu)
- Include timestamp
- Use top-level key "d" for all addresses

4. Go to the Address tab and set the addresses for reported status (LB-0) and desired setting (LB-1). ->, <-> stands for the direction in which data is transmitted.

The screenshot shows a dialog box titled "Address setting" with a close button (X) in the top right corner. It has an "Advanced mode" checkbox which is currently unchecked. The "Name" field contains "default1". The "Type" is set to "Bit". There are two checked checkboxes for status and setting:

- Status (Device address -> AWS IoT "reported")
  - Device: Local HMI (dropdown)
  - Address: LB (dropdown) 0 (text box)
- Setting (Device address <-> AWS IoT "desired")
  - Device: Local HMI (dropdown)
  - Address: LB (dropdown) 1 (text box)

At the bottom, there is an unchecked checkbox:  Remove JSON array bracket '[' and ']'

5. In the Advanced Mode settings window, Status (reported) and Setting (desired) can use different addresses, and data is transmitted to/from AWS IoT/device.

Address setting ✕

Advanced mode

Name:

Type:

Status (Device address -> AWS IoT "reported")

Send initial value when HMI starts

Device:  ⏪ ⏩

Address:

Status (AWS IoT "reported" -> Device address)

Device:  ⏪ ⏩

Address:

Setting (Device address -> AWS IoT "desired")

Send initial value when HMI starts

Device:  ⏪ ⏩

Address:

Setting (AWS IoT "desired" -> Device address)

Device:  ⏪ ⏩

Address:

---

Remove JSON array bracket '[' and ']'

Your industrial control solutions source

[www.maplesystems.com](http://www.maplesystems.com)

