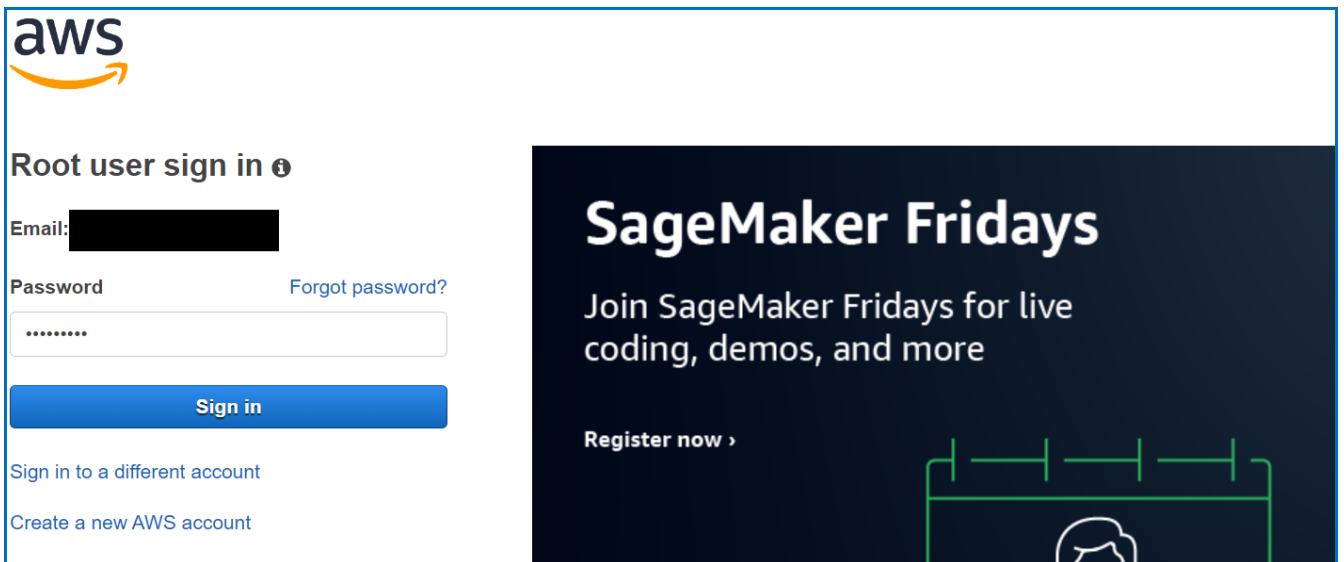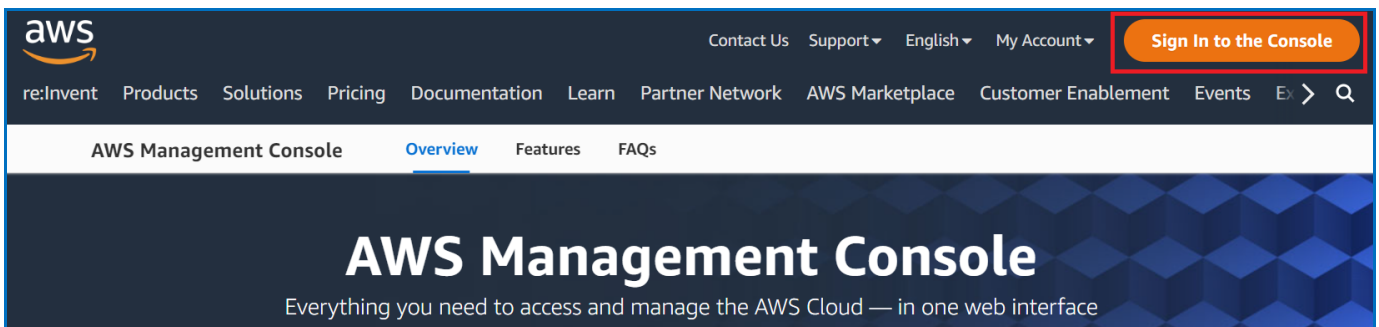## FAQ-CNV-08: UA Web UI Function Wizard – Module Communication Conversion - How do UA series upload data to AWS?

This FAQ introduces the UA series to use MQTT to upload Modbus data to AWS.
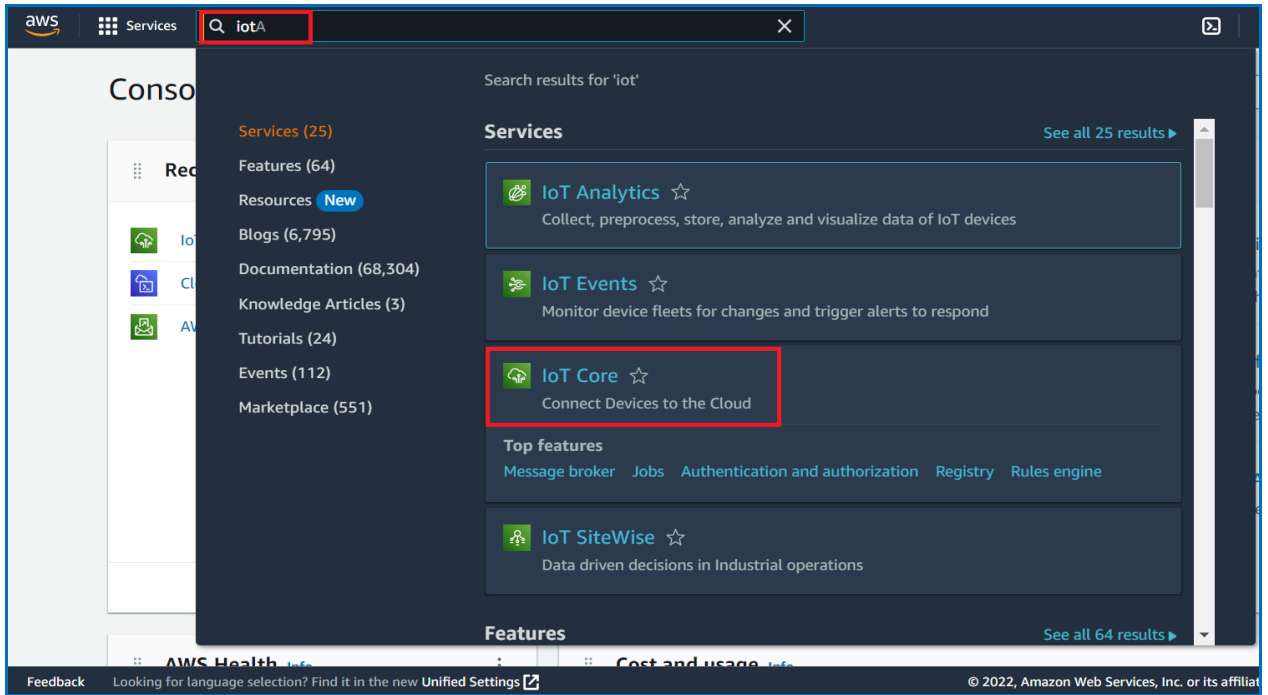
Set up the AWS environment

1. Log in to AWS and AWS IOT
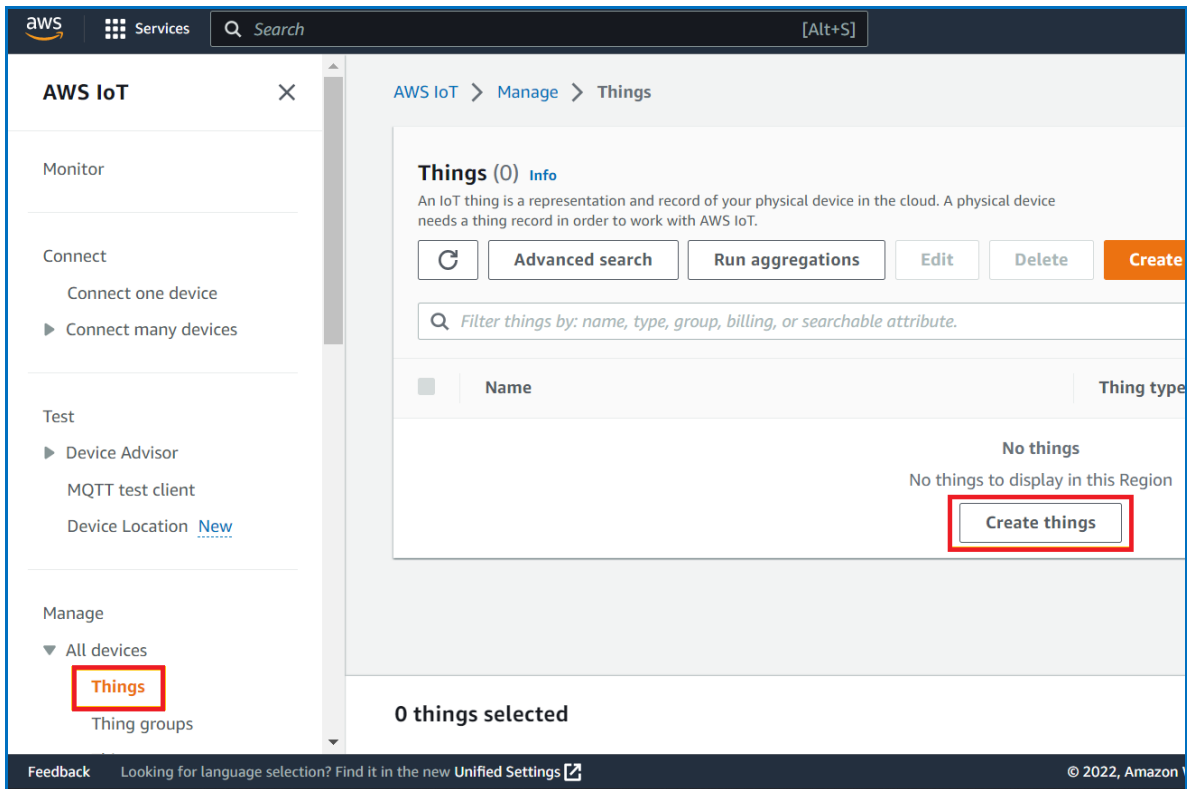
Use a browser to log in to the AWS website.

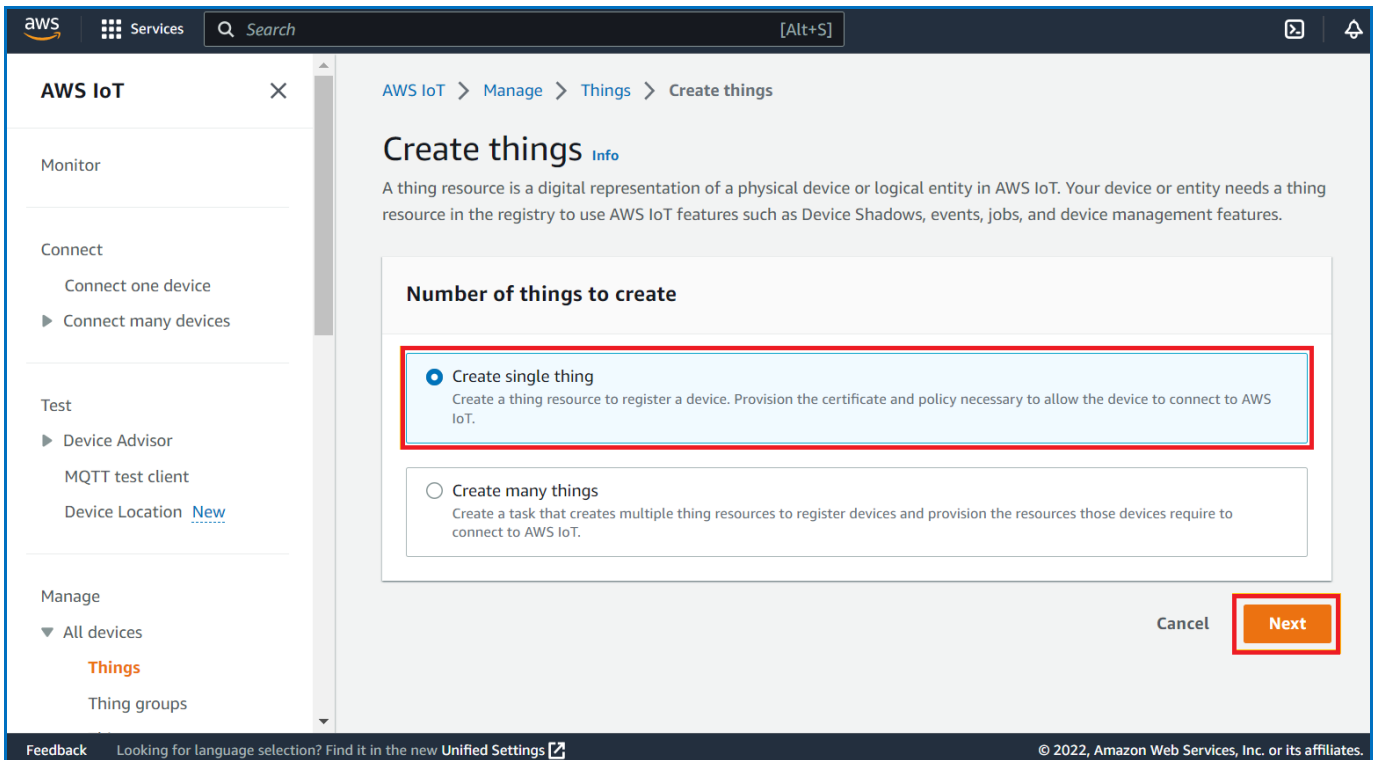Fill in "**iot**" in the search box of AWS Service, select "**IoT Core**" service.

2. Create Device Things

On AWS IoT Service, select "Manage -> **Things**" on the left, and click "**Create things**" to create the IoT device.

Fill in the device name in the "**Thing name"** field, for example: UA-5231.
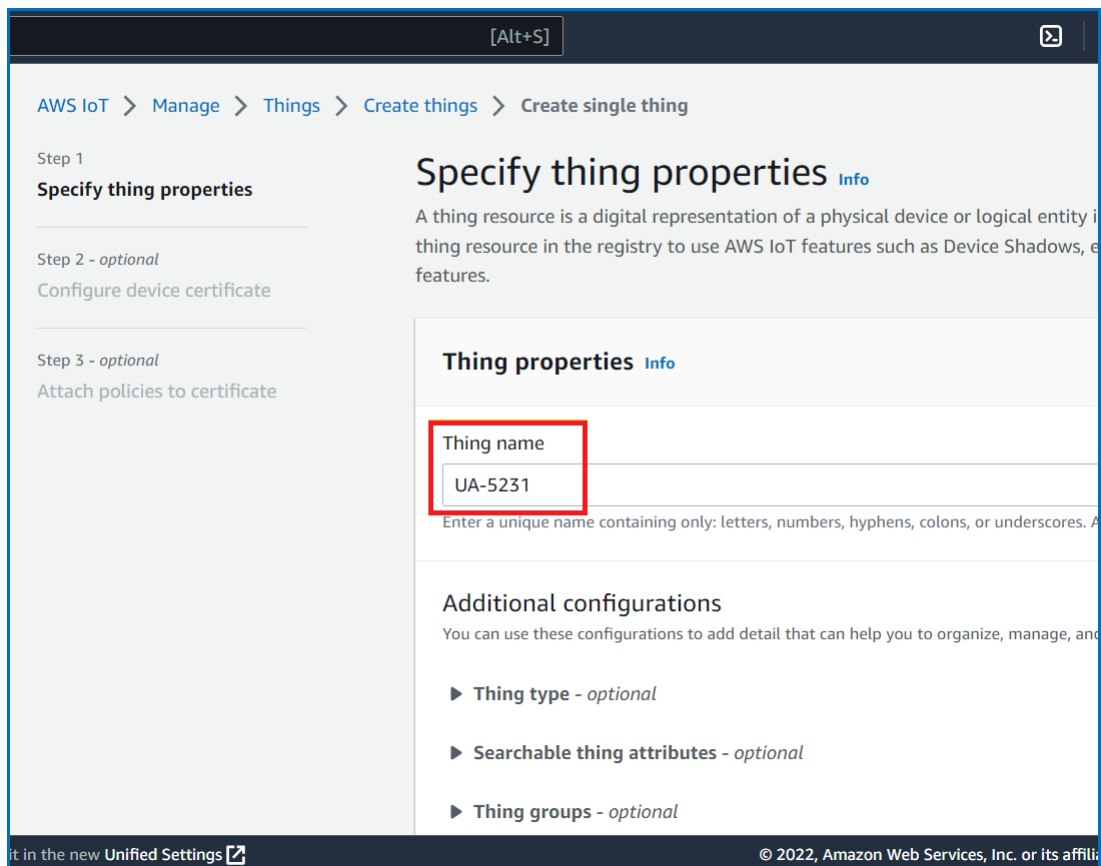
Then go to the bottom of the page and select **Next**.

Activate the certificate and link to the device.

Download all Certificates and Key files to the local PC for storage, and then click **Done** to complete.

3. Create an AWS IoT Policy and link to the certificate

Click "**Security** -> **Policies**" to **create** a Policy.
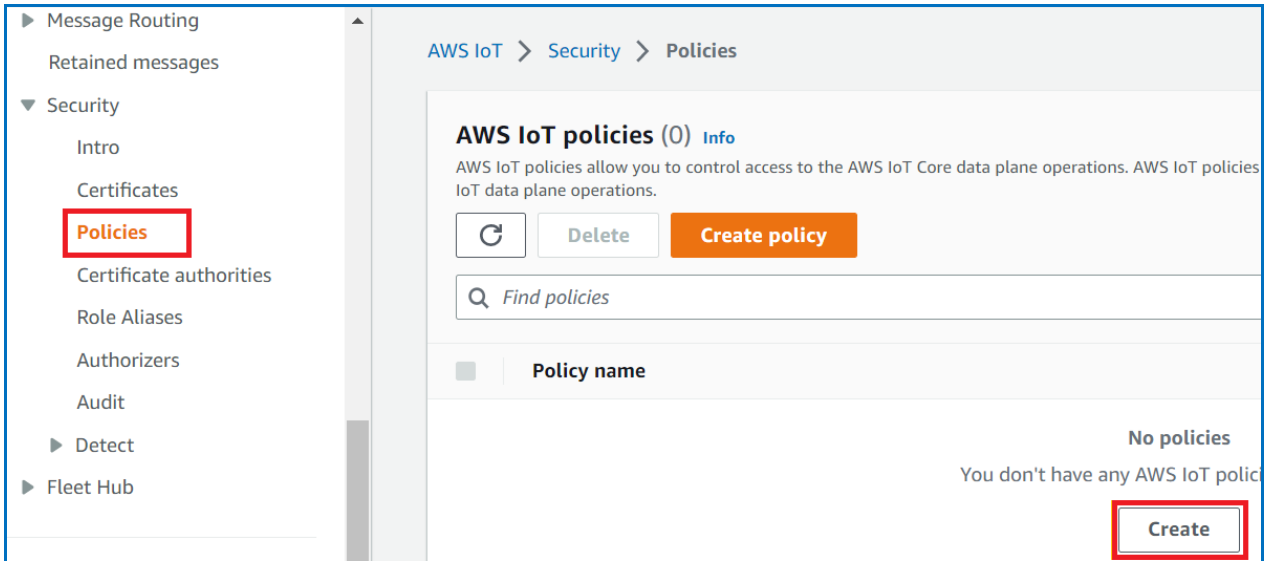
Set up access permissions and device sources. You can refer to the figure below and set it as no restriction.

After settings, you can view the Policy in the list.

Click "Security -> Certificates" to create an "Attach policies".

Select the Policy.

Completing the setting is shown in the figure below.

Set up the UA series MQTT

Please check if your UA product is connected to the router/Internet and powered on.

● Create **MQTT Connection**

Select an MQTT Function Wizard to create a Modbus (or other) / MQTT project (please choose according to your project needs).

1. Modbus Module Setting

2. MQTT Certificate Setting

    (a) Upload the **AmazonRootCA3.pem** downloaded from AWS IoT to UA via the **Trusted Certificate** item.

    (b) Upload the **xxxxxxxxxx-certificate.pem.crt** downloaded from AWS IoT to UA via the **Certificate** item.

(c) Upload the **xxxxxxxxxx-private.pem.key** downloaded from AWS IoT to UA via the **Private Key** item.

MQTT Client

Remove the File

Trusted Certificate	Remove	
Certificate	Remove	
Private Key	Remove	

Upload the file to the controller

Trusted Certificate	Select File	Upload	Success.
Certificate	Select File	Upload	Success.
Private Key	Select File 8e2e9d7fd-private.pem.key	Upload	

3. MQTT Broker Setting

(a) In MQTT Connection, set up a Remote Broker.

Module Setting ➤ MQTT Certificate ➤ MQTT Broker Setting ➤ Enable Converting Module ➤ Save Project ➤

Run the project

System Setting   Module Setting   **IoT Platform Setting**   Convert Setting   Advanced Setting   Logger Setting

I/O Status   File Setting

IoT Platform Setting ⟩ Remote Broker

MQTT Connection

Local Broker

Remote Broker ⬅

MQTT Group Connection

Microsoft Azure Platform

OPC UA Connection

Local Server

Remote Broker List

Remove ☐	Broker Name	IP / Domain	Port
⊕	Name1		
☐	AWS	127.0.0.1	1883

Remove		< 1 / 1 >

Save

(b) In **the Broker Content Settings** > **IP/Domain** item, fill in the **Endpoint** item data on the **AWS IoT** > **Settings** page.

(c) Broker Content Settings > **Port**: 8883

(d) Broker Content Settings > **SSL/TLS**: check Enabled
Trusted Certificate, Certificate and Private Key: select previously uploaded files in order.

(e) Anonymous Login: check Enabled

4. Enable Converting Module

After completing Broker settings, select the Modbus device Tag to be converted. (Message QoS set to 0 or 1)

MQTT Client Setting

No.	1
Module Name	simulate
Scan Rate(ms)	1000
Dead Band	0
Will Topic	
Will	
MQTT Connection	☐ Broker (Local) ☑ AWS (Remote)

Publish & Subscribe

Details [Show] [Hide]

Name	Subscribe Topic	Subscribe QoS	Publish Topic	Publish QoS	Retain ☐	Enabled ☑
Tag0	/MTCP_No.1_simulate/Holding_Registers/Tag0/Subscribe	1 ⌄	/MTCP_No.1_simulate/Holding_Registers/Tag0/Publish	1 ⌄	☐	☑
Tag1	/MTCP_No.1_simulate/Holding_Registers/Tag1/Subscribe	1 ⌄	/MTCP_No.1_simulate/Holding_Registers/Tag1/Publish	1 ⌄	☐	☑
Tag2	/MTCP_No.1_simulate/Holding_Registers/Tag2/Subscribe	1 ⌄	/MTCP_No.1_simulate/Holding_Registers/Tag2/Publish	1 ⌄	☐	☑

5. Save the project and run the project
After completing the preceding steps, save and run the project following the wizard steps.

Test

1. Please click the **MQTT test client** on the **AWS IoT** service page, and use "**Subscribe to a topic**" to check whether the data has been uploaded successfully.

2. Click **MQTT test client** on the AWS IoT service page, and use "**Publish to a topic**" to check whether the **UA** series has successfully received the data.