

# 1 Demo – AWS IoT Platform

## 1.1 Modbus RTU Device Interface:

### Device information:

RS485 interface setting:

Baud Rate	Word Length	Parity	Stop Bits
9600	8	none	1

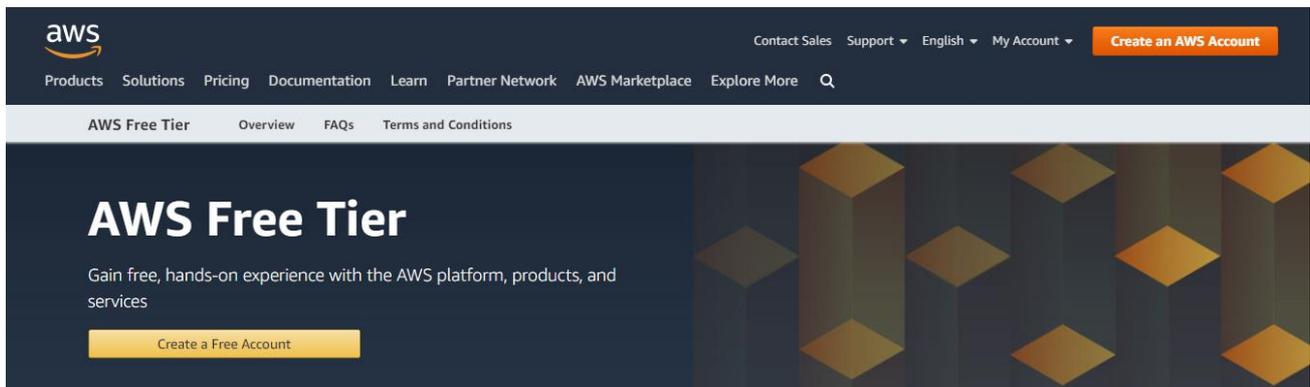
Modbus register information:

	Modbus ID	Modbus Fcode	Address Data	Length Data	Type
CO2	0x1	0x3	0x0000	0x0001	Decimal

## 1.2 Set up AWS IoT Platform Server

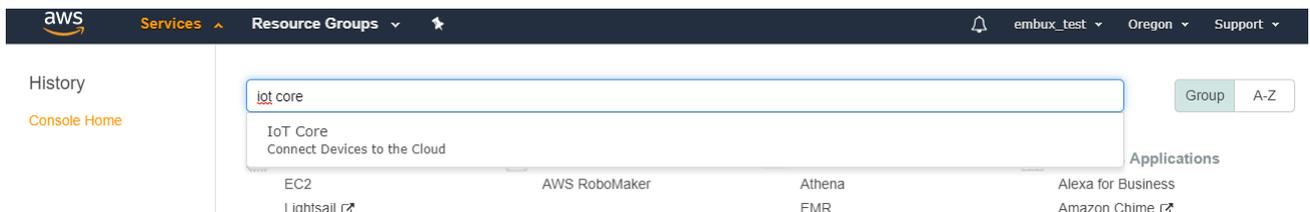
### a. Create an Amazon account:

Get into <https://aws.amazon.com/tw/> and create an account.

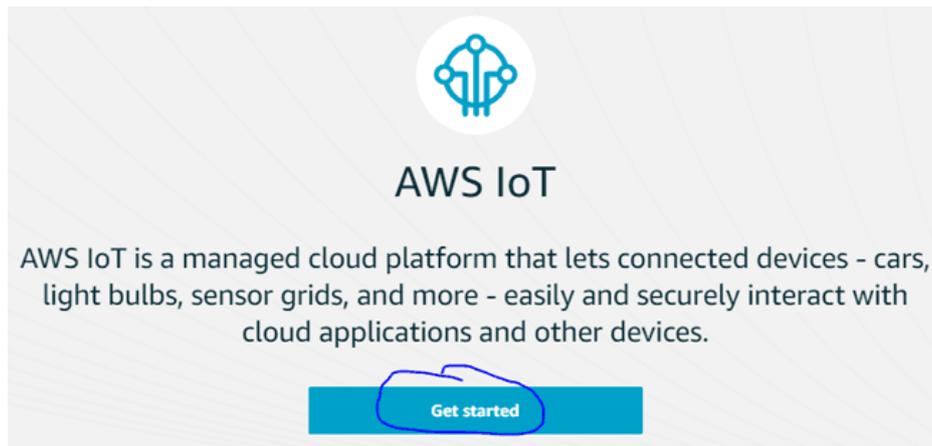


### b. Create an AWS Thing with Certificate and Policy:

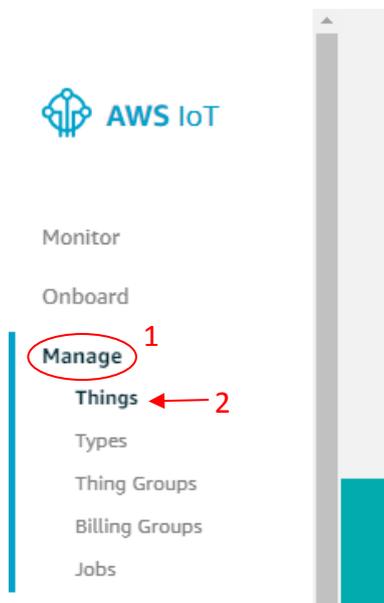
- Use AWS services to search for "IoT Core". It's being listed as shown below. Click on it to open the AWS IOT console.



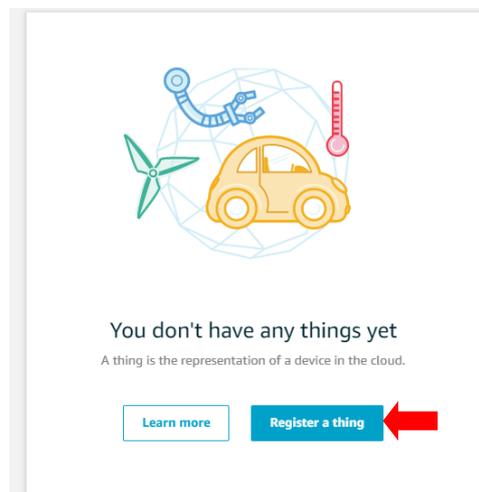
There's an introductory message from AWS IoT, just click on "get started".



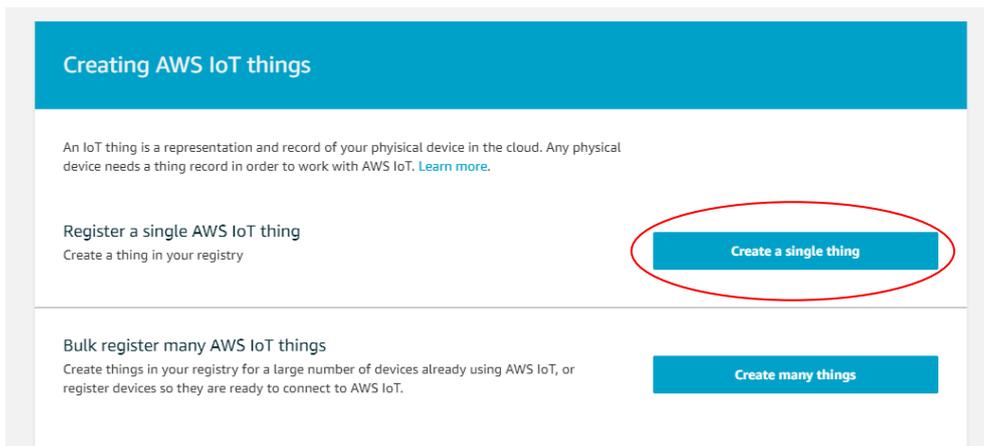
ii. Select "Manage" and "Things" at the left side of the screen.



iii. To create the thing, just click on the "Register a thing".



iv. Click on “create a single thing”.



**Creating AWS IoT things**

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

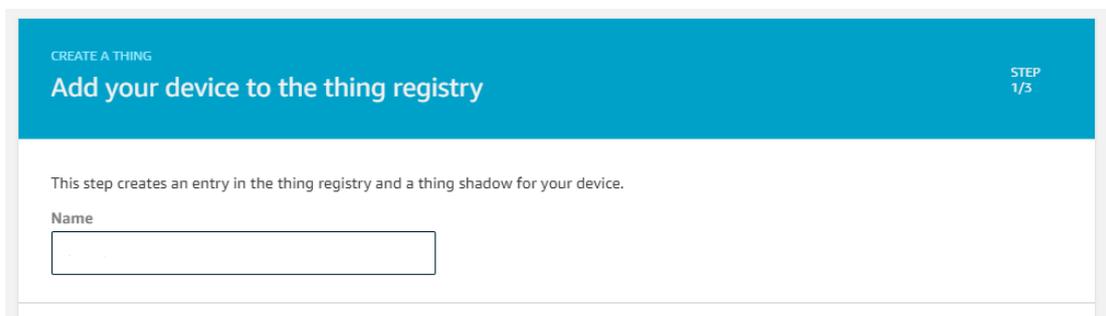
**Register a single AWS IoT thing**  
Create a thing in your registry

**Create a single thing**

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

**Create many things**

v. Provide a name which can be anything. After naming, just scroll down and click on “next”.



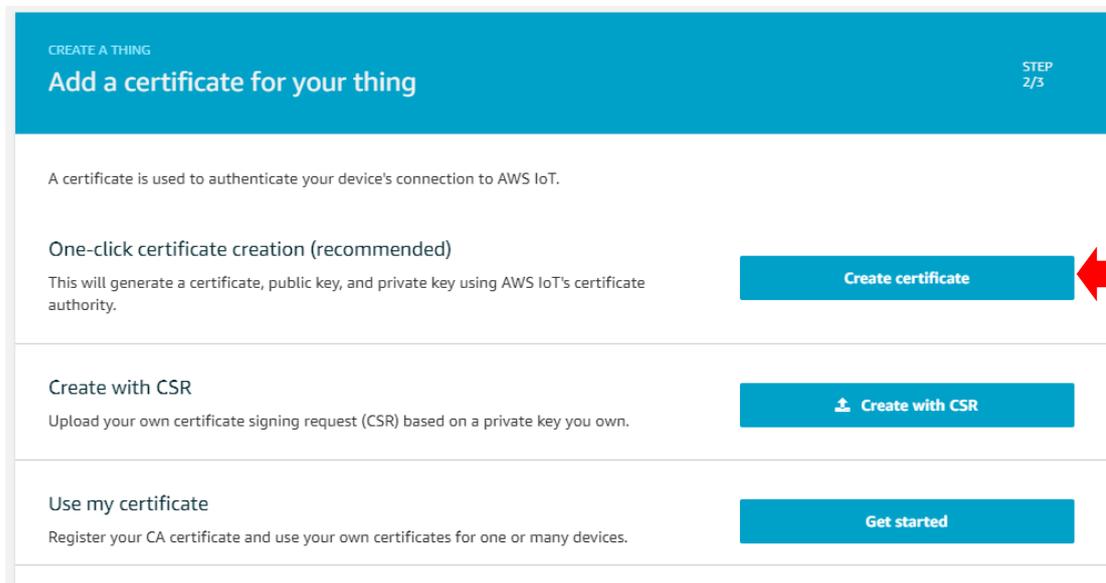
**CREATE A THING** STEP 1/3

### Add your device to the thing registry

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

Name

vi. Click “Create Certificate”.



**CREATE A THING** STEP 2/3

### Add a certificate for your thing

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

**One-click certificate creation (recommended)**  
This will generate a certificate, public key, and private key using AWS IoT's certificate authority.

**Create certificate**

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

**Create with CSR**

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

**Get started**

- vii. Download the three key files and save it on the computer somewhere secure, and click on “Attach a policy”.

Make sure you click on the Activate button first.

## Certificate created!

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

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

A certificate for this thing	ef0f35d28e.cert.pem	<a href="#">Download</a>	←
A public key	ef0f35d28e.public.key	<a href="#">Download</a>	
A private key	ef0f35d28e.private.key	<a href="#">Download</a>	←

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

A root CA for AWS IoT [Download](#) ←

[Activate](#) ←

Cancel [Done](#) [Attach a policy](#)

For some users the CA file might open as a stream of code on chrome. In that case, just save it. Make sure of changing the extension of the file to .pem if it ends with .text.

- viii. Click “Register Thing”.

The policy will be created in next step and then attach it.

## CREATE A THING STEP 3/3

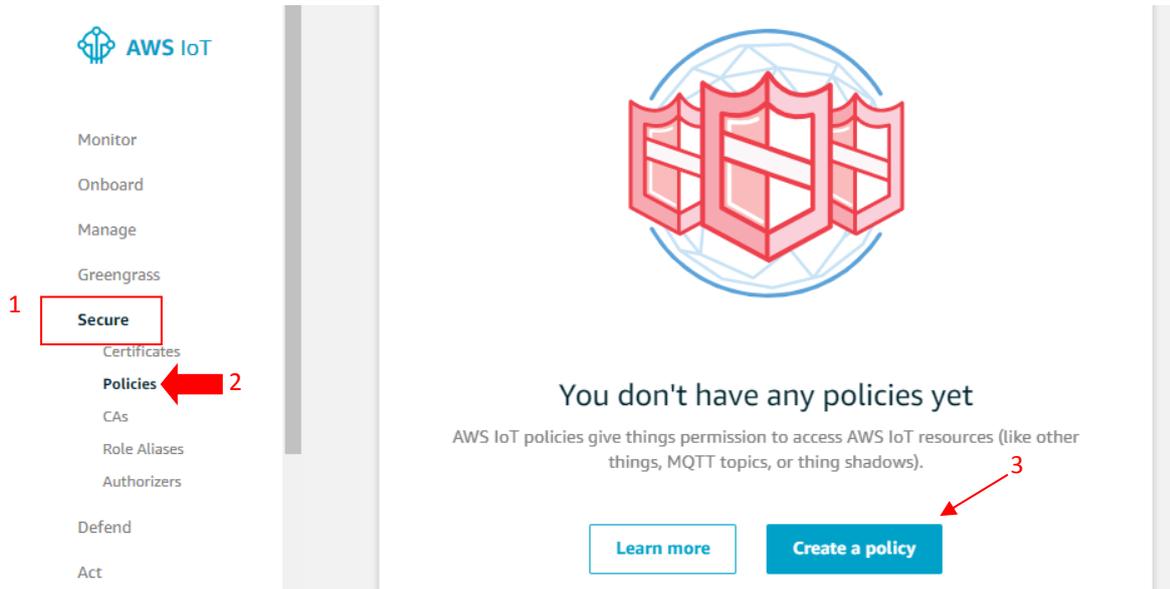
### Add a policy for your thing

Select a policy to attach to this certificate:

**No match found**  
There are no policies in your account.

0 policies selected [Register Thing](#)

- ix. Click “Secure” → “Policies” on the left side menu, then create a policy.



- x. Fill out the form, then click “Create” on the end of the page.

**Create a policy**

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

Name

---

**Add statements** Advanced mode

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

**Action**

Please use commas to separate actions. e.g. iot:Publish, iot:Subscribe

**Resource ARN**

Specific resources must include client ID ARN, topic ARN, or topic filter ARN.

**Effect**

Allow  Deny Remove

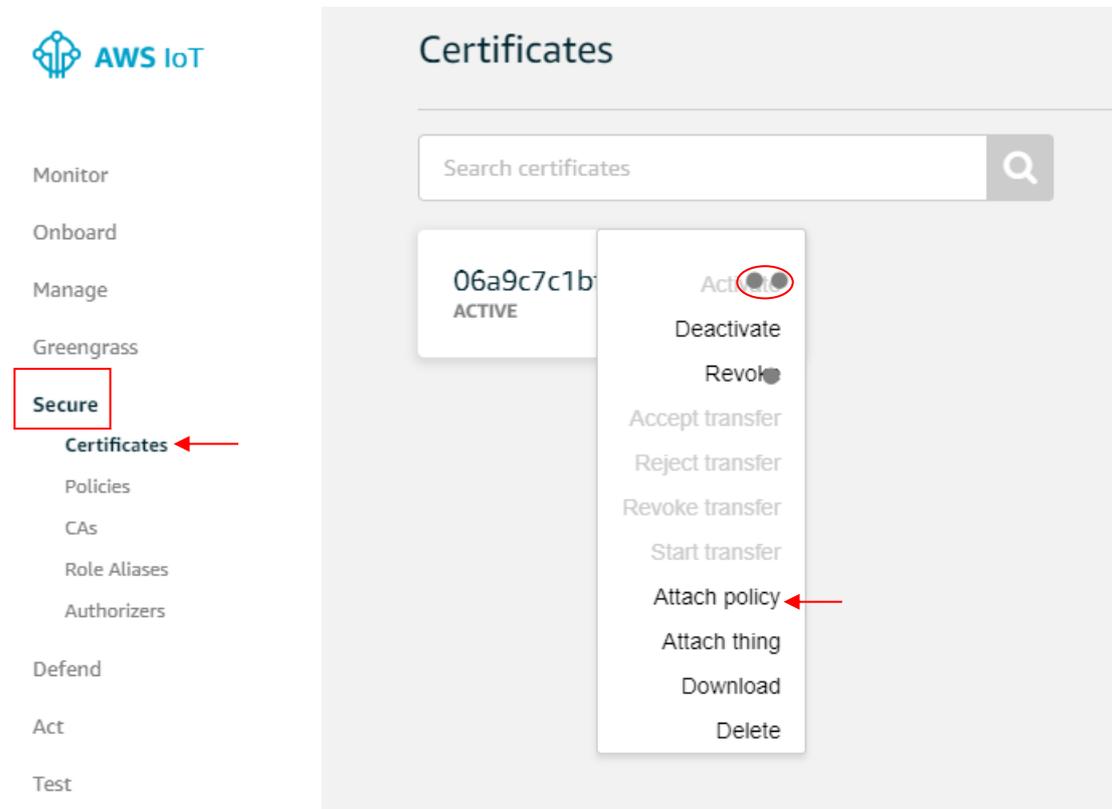
Action	iot:Publish, iot:Subscribe, iot:Connect, iot:Receive, iot:GetThingShadow, iot:UpdateThingsShadow
Resources ARN	arn:aws:iot:region:AWS account ID:resource type/resource name

For more information, please refer to:

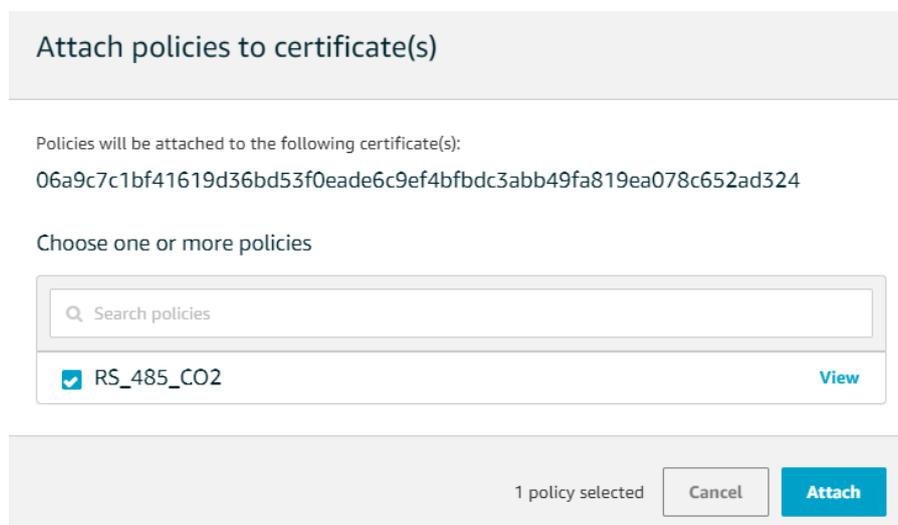
<https://docs.aws.amazon.com/iot/latest/developerguide/iot-policies.html>

Make sure the allow button is checked.

- xi. Select secure and certificates, clicking on options and choosing “Attach policy”.



- xii. Select the policy which was just created and then click on the attach button.



### c. Getting your AWS thing Details:

Broker address can be found by clicking on the name of the thing in manage option.

THING  
**RS\_485\_C02**  
NO TYPE

Details This thing already appears to be connected.

Security

Thing Groups HTTPS

Billing Groups Update your Thing Shadow using this Rest API Endpoint. [Learn more](#)

Shadow

**Interact** [Redacted] iot.us-west-2.amazonaws.com

## 1.3 Setup NIO51

### a. Setup Modbus to MQTT

Open NIO51 web, default IP is <http://192.168.1.1>, and go to NIO-IOT application setting page. (*username: root, password: admin*)

NEXCOM NIO51

**Authorization Required**  
Please enter your username and password.

Username

Password

Powered by LuCI / NIO51 / v1.1.97

Click NIO-IOT, then step by step to setup Modbus to MQTT.

NEXCOM NIO51 Status System Network NIO-IOT Logout

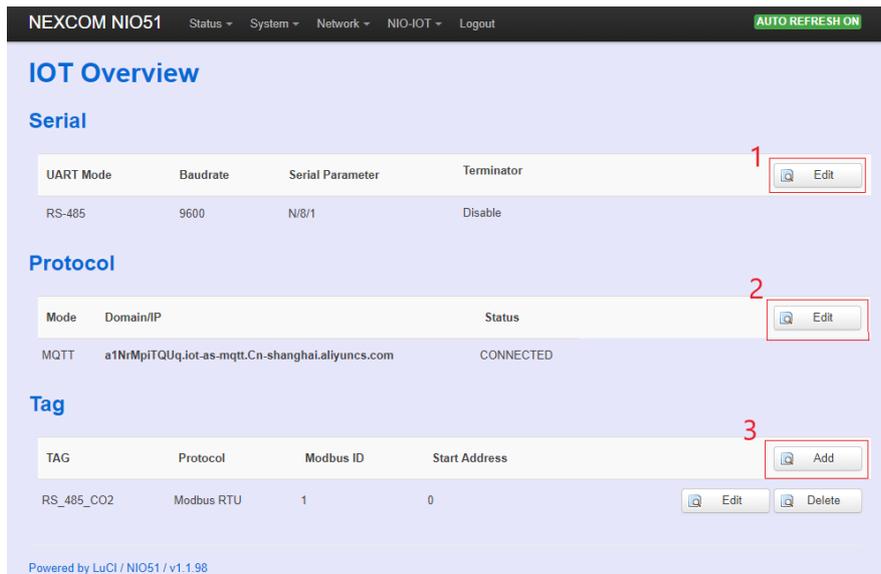
**Status**

System

Hostname	NIO51
Model	NIO51
Firmware Version	NIO51-v1.1.97 / LuCI (git-15.216.69575-bb7ea3e)

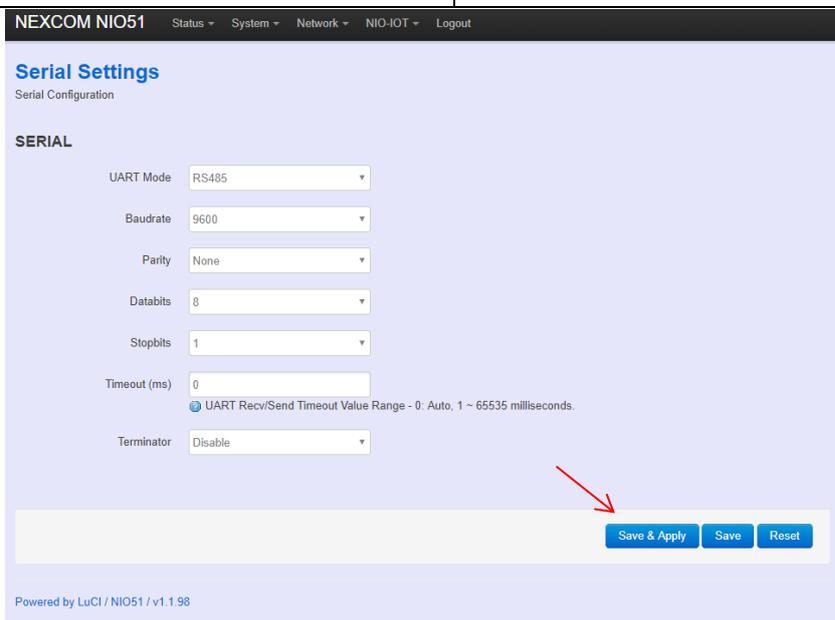
**NIO-IOT**

- JSON Format
- CA Certificate
- Modbus Log
- MQTT Log



i. Edit “serial” setup to follow the target device spec.

UART Mode	RS485
Baudrate	9600
Parity	None
Databits	8
Stopbits	1



Please click “Save&Apply” after every editing.

ii. Edit Protocol.

Protocol Mode	Modbus to MQTT
Broker Domain/IP	almqj83mdd5cq-ats.iot.us-west-2.amazonaws.com
Broker Port	8883
SSL/TLS Encryption	Enable

NEXCOM NIO51 Status System Network NIO-IOT Logout

### Protocol Settings

Protocol Configuration

IOT Settings

Protocol Mode: Modbus to MQTT

Client ID:    
  Leave blank to use random Client ID

Broker Domain/IP: almql83mdd5cq-ats.iot.us-wes

Broker Port: 8883

Keep Alive: 60   
  Defines the longest period of time that the broker and client can endure without sending a message

SSL/TLS Encryption: Enable

Anonymous Login: Enable   
  Connect without using username and password. (Server must enable anonymous login)

Scan Rate(s): 10   
  Time range between publish : 1 ~ 65535 seconds.

Clean Data After Restart:

Send JSON Format:

Save & Apply Save Reset

### iii. Edit Tag

Tag Name	RS_485_CO2
Modbus Protocol	RTU
Modbus ID	1
Modbus Function	3
Start Address	0
Data Type	UNIT 16
Data Number	1

Publish topic is a user decision.

NEXCOM NIO51 Status System Network NIO-IOT Logout

### MQTT Settings

MQTT Configuration

Tag Settings

Tag Name: RS\_485\_CO2

Modbus Protocol: RTU

Modbus ID: 1

Modbus Function: 3:Read Holding Registers

Start Address: 0

Data Type: UINT16

Data Number: 1

SWAP:

Publish Topic: /a1NrMpITQUg/NIO\_51/user/   
  Limited to 128 character

Subscribe Topic:    
  Limited to 128 character

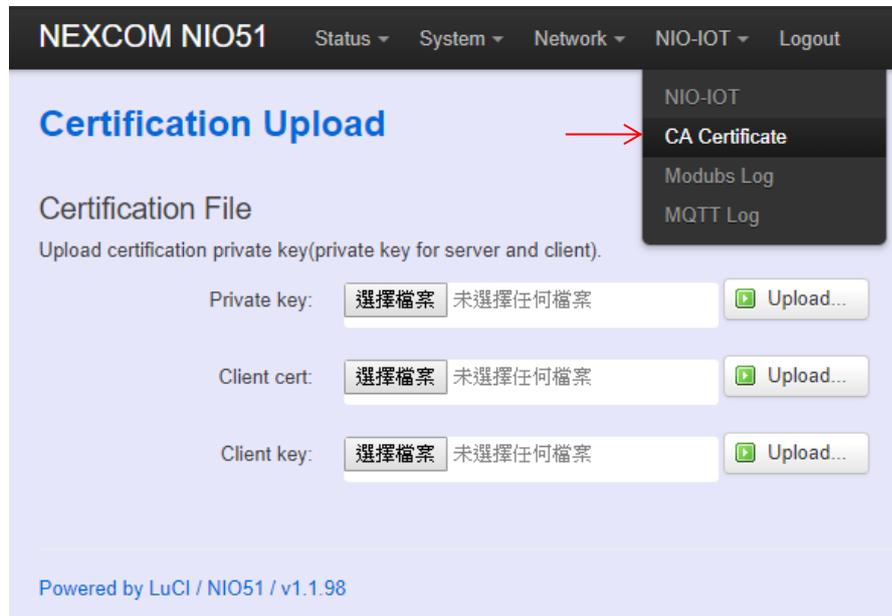
Qos: 0 (at most once)   
  Quality of Service Levels

Save & Apply Save Reset

#### iv. Upload Certification.

Upload certification private key.

Click NIO-IOT, then click CA Certificate.



Private key	AmazonRootCA1.pem
Client cert	XXXXXXXXXX -certificate.pem.crt
Client key	XXXXXXXXXX -private.pem.key

According to 1.2-a-vii, the Certification File should be saved somewhere in the computer by user.

## 1.4 Verification

