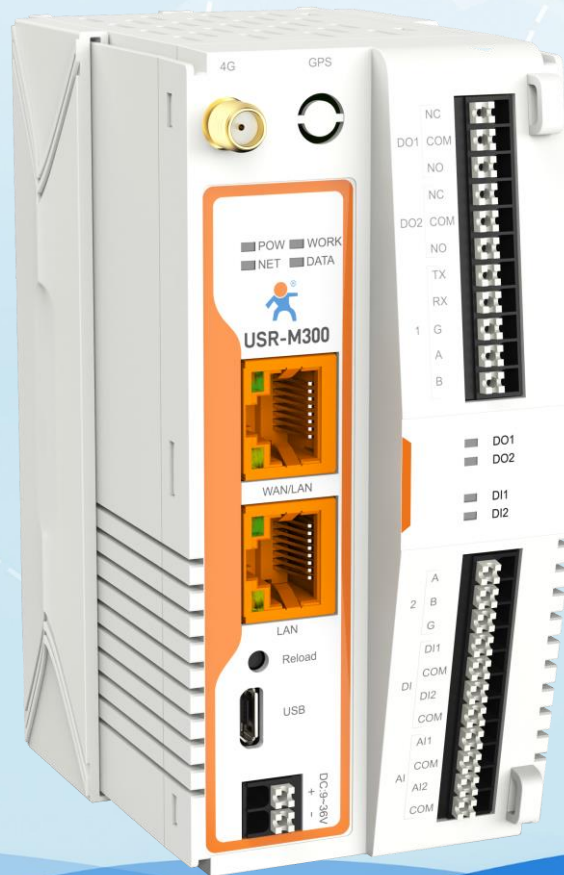


USR-M300 Quick Start Guide with AWS IoT



Be Honest & Do Best

Your Trustworthy Smart Industrial IoT Partner

- 1. Introduction..... - 3 -
- 2. AWS IoT..... - 3 -
 - 2.1. Basic Parameters..... - 3 -
 - 2.2. Publish Settings - 4 -
 - 2.3. Subscribe Settings..... - 4 -
- 3. AWS Connection Test - 4 -
 - 3.1. Preparations - 4 -
 - 3.2. Configuration of AWS..... - 4 -
 - 3.2.1. Login..... - 4 -
 - 3.2.2. Create Things and Certificates - 5 -
 - 3.2.3. Create Policies - 9 -
 - 3.2.4. Attach Polices to Certificate - 10 -
 - 3.2.5. Obtain Product Information - 11 -
 - 3.3. Device Configuration - 12 -
 - 3.4. Device Activity..... - 14 -
 - 3.5. Data Transmission Test - 14 -

1. Introduction

USR-M300 is a high-performance and scalable edge IOT gateway. This device integrates edge collection, data calculation, data reading and writing, active reporting, linkage control, IO collection and control and other functions. The collection protocol includes standard Modbus protocol and a variety of common PLC protocols, as well as industry-specific protocols. At the same time, the product also has routing and VPN as well as graphical programming functions to ensure data transmission security. Using graphical programming, users can develop independently to achieve the required functions.

USR-M300 is embedded in Linux kernel, with a main frequency of up to 1.2Ghz. It can access the Internet via Ethernet port, ADSL and LTE cat4 cellular network to achieve easy network deployment.

It is widely used in various industrial intelligent solutions such as industrial robot, smart factories, smart agriculture, smart water management system etc.

2. AWS IoT

USR-M300 supports connecting to AWS IoT platform via MQTT, which can be achieved via simple MQTT and SSL parameter configuration. At the same time, the device's edge computing and AWS functions support combined configuration, which is very flexible and practical.

2.1. Basic Parameters

- Protocol Select: MQTT version, supports MQTT-3.1 and MQTT 3.1.1.
- Client ID: MQTT client identifier.
- Remote Server Address: MQTT server domain name or IP address.
- Remote Port: MQTT server port.
- Heartbeat time: MQTT protocol heartbeat time.
- Reconnection Interval: The interval between the current connection failure and the next MQTT connection.
- Connection Verification: When enabled, after connecting to the MQTT server, the username and password will be sent for connection authentication.
- Username: MQTT connection username, used for connection authentication.
- Password: MQTT connection password, used for connection authentication.
- Last Will: MQTT connection flag. When the network connection is closed, the server must publish this will message.

- SSL protocol: Supports TLS1.0 and TLS1.2 versions, and the authentication mode can choose none certificate authentication, CA signed server and self signed certificate.

2.2. Publish Settings

- Topic: Publish topic name.
- QOS: Message quality of published topics.
- Message retained: MQTT retains the message flag, which is used by the server to store this application message and its quality of service (QoS).

2.3. Subscribe Settings

- Topic: Subscribe topics.
- QOS: Message quality of subscribed topics.

3. AWS Connection Test

In this case, we will show how to connect M300 to AWS.

3.1. Preparations

- USR-M300*1
- RS485 serial to USB cable*1
- Ethernet cable*1
- 12V/1A power adaptor*1

3.2. Configuration of AWS

3.2.1. Login

1. Login <https://www.amazonaws.cn/en/>
2. Login to the account. If you do not have an account, please create one firstly.

Sign in as IAM user

Account ID (12 digits) or account alias

IAM user name

Password [Forgot password?](#)

Remember this account

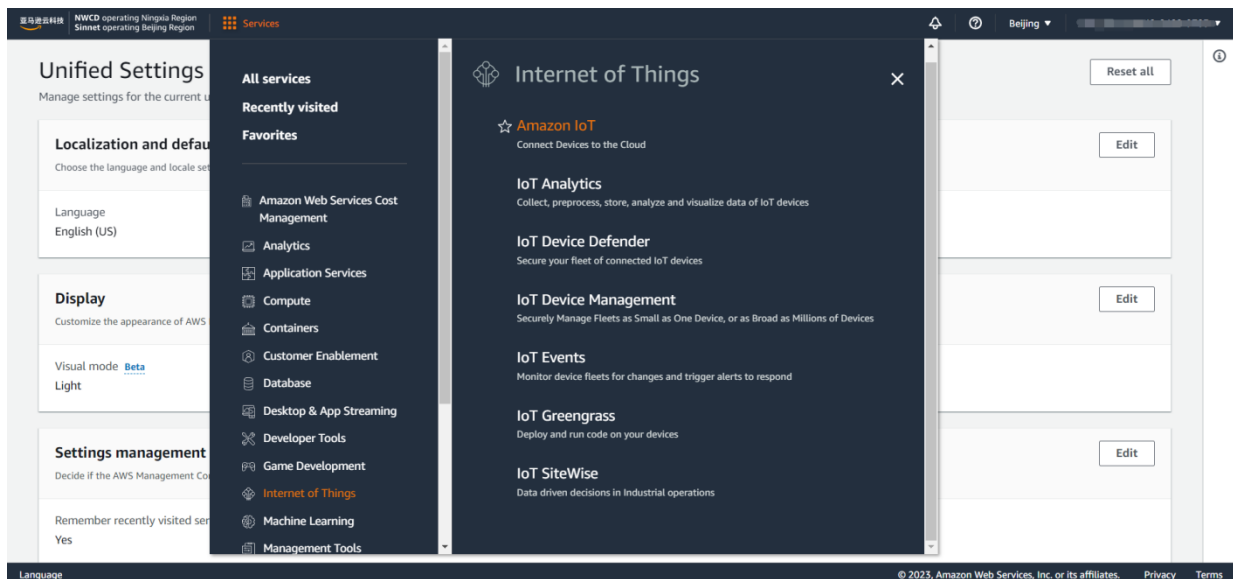
Sign in

[New to Amazon Web Services?](#)

[Create a new Amazon Web Services account](#)

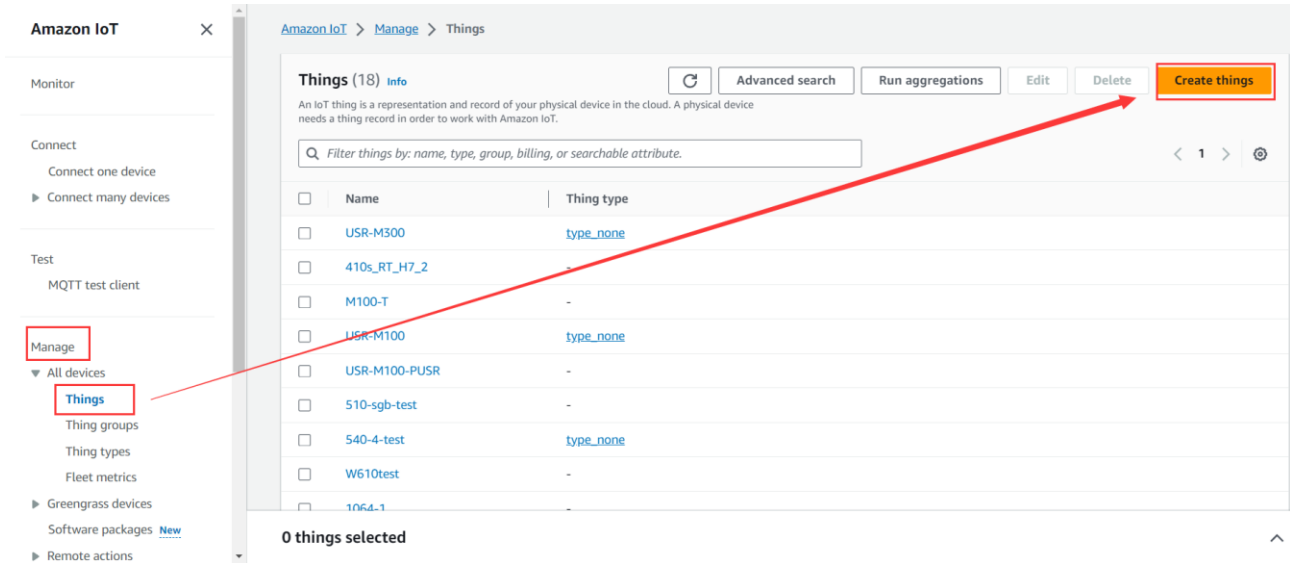


3. Find **Amazon IoT** in **Internet of Things**.

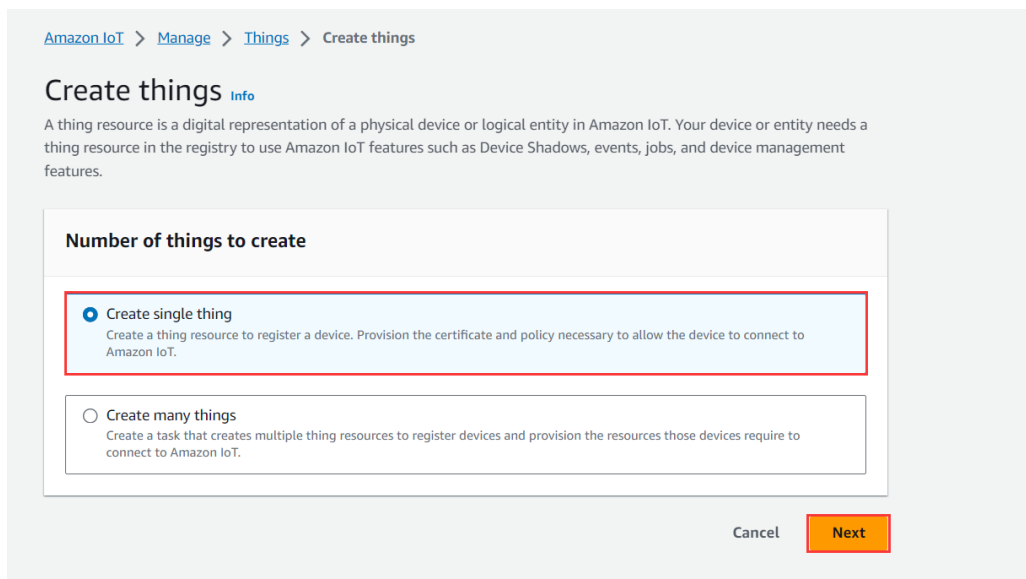


3.2.2. Create Things and Certificates

1. In **Manage->All devices->Things**, click **Create things** to add the device.



2. Choose **Create signal thing**, fill in the **Thing name** and **Thing type**, here we configure the thing name to “USR-M300-Test”, configure the thing type to “type_none”.



Amazon 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 Amazon IoT. Your device or entity needs a thing resource in the registry to use Amazon IoT features such as Device Shadows, events, jobs, and device management features.

Thing properties [Info](#)

Thing name

USR-M300-Test

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

Thing types are an optional way to store description and configuration information that is common to things that have the same thing type.

Thing type

type_none ▼ Clear Create thing type

Add searchable attributes to allow your thing to be grouped and searched without using fleet indexing.

No searchable attributes are associated with the selected thing type.

Add new attribute

- ▶ Non-searchable thing attributes - optional
- ▶ Thing groups - optional
- ▶ Billing group - optional
- ▶ Packages and versions - optional

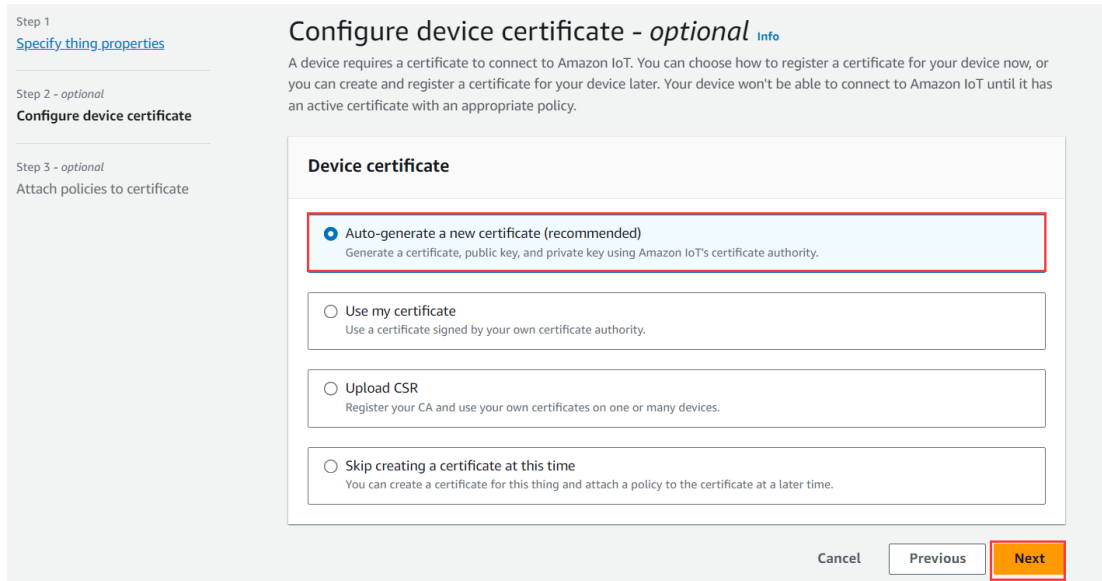
Device Shadow [Info](#)

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

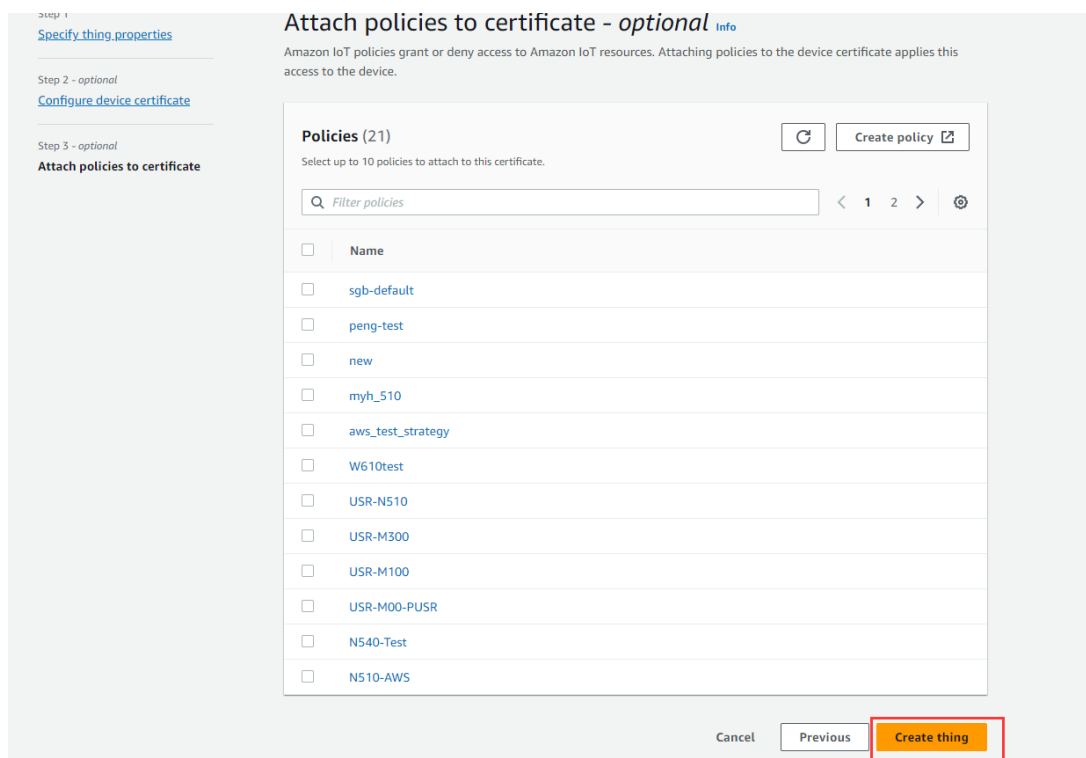
- No shadow
- Named shadow
Create multiple shadows with different names to manage access to properties, and logically group your devices properties.
- Unnamed shadow (classic)
A thing can have only one unnamed shadow.

Cancel **Next**

- Next, choose **Auto-generate a new certificate**, you can also choose others if you have your own certificates. Then click **Next**.



- Do not create policies, directly click **Create thing**. Then it will show the certificates interface. Download the certificate and key files, then click **Done** to back to the things interface.



Download certificates and keys ✕

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

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

Device certificate Deactivate certificate Download

b5160b9bfa2...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

b5160b9bfa2e472c6b63bea...0592a85-public.pem.key

Private key file Download

b5160b9bfa2e472c6b63bea...592a85-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, Amazon IoT supports additional root CA certificates. These root CA certificates and others are available in our developer guides. [Learn more](#)

Done

3.2.3. Create Policies

1. In **Manage->Security->Policies** interface, click **Create policy**.

The screenshot shows the Amazon IoT console interface. On the left, the 'Manage' menu is expanded to show 'Security', which is further expanded to show 'Policies'. The main content area displays 'Amazon IoT policies (21)' with a search bar and a 'Create policy' button highlighted by a red arrow. A list of existing policies is visible, including W610test, USR-N510, USR-M300, USR-M100, USR-M00-PUSR, sgb-default, peng-test, new, N540-Test, and N510-AWS.

2. Fill in the **Policy name**, add new statement in **Policy document**. There is a default statement which can be directly operated.
3. Add 4 policies: iot:Connect, iot:Publish, iot:Receive, iot:Subscribe.
4. Policy resource format: arn:aws:iot:region:AWS-account-ID:Resource-type/Resource-name, AWS-account-ID is your account ID of AWS.
5. Then click **Create** to create the policies.

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

Policy name
 USR-M300-Test

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 statements | Policy examples

Policy document Info Builder JSON

An Amazon 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.

Policy effect	Policy action	Policy resource	
Allow	iot:Connect	arn:aws:iot:cn-north-1:944284229783:client/*	Remove
Allow	iot:Publish	arn:aws:iot:cn-north-1:944284229783:topic/*	Remove
Allow	iot:Receive	arn:aws:iot:cn-north-1:944284229783:topic/*	Remove
Allow	iot:Subscribe	arn:aws:iot:cn-north-1:944284229783:topicfilter/*	Remove

Add new statement

3.2.4. Attach Polices to Certificate

- We have bound certificate and the thing when creating the device, so we can directly find the device in **Manage->All devices->Things**, click **Certificates** in **USR-M300-Test**.

Connect many devices

Test
 MQTT test client

Manage

All devices

Things

Thing groups

Thing types

Fleet metrics

Greenrass devices

Software packages New

Remote actions

Message routing

Retained messages

Security

Device software

Billing groups

USR-M300-Test Info Create secure tunnel Edit Delete

Thing details

Name
 USR-M300-Test

Type
 type_none

ARN
 arn:aws-cn:iot:cn-north-1:944284229783:thing/USR-M300-Test

Attributes **Certificates** Thing groups Device Shadows Activity Packages and versions Jobs Alarms Defender metrics

Certificates (1) Info Refresh Detach Create certificate

The device certificates attached to this thing resource.

Find certificates

Certificate ID	Status
b5160b9bfa2e472c6b63bea35efba45348c82f024048782c458e4bdbd0592...	Active

- Find **Polices** under certificate, click **Attach policies**, choose the polices you have created.

Policies (0) Info

Amazon IoT policies allow you to control access to the Amazon IoT Core data plane operations.

Detach policies Attach policies

No policies
You don't have any policies attached to this certificate.

Attach policies to the certificate

b5160b9bfa2e472c6b63bea35efba45348c82f024048782c458e4bdb0592a85.

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

Choose Amazon IoT policy

USR-M300-Test

Cancel Attach policies

3.2.5. Obtain Product Information

- Client ID is the Things name of the device, you can find the device name in **Manage->All devices->Devices**.

Connect many devices

Test
MQTT test client

Manage

- All devices
 - Things**
 - Thing groups
 - Thing types
 - Fleet metrics
- Greengrass devices
- Software packages **New**
- Remote actions
- Message routing
- Retained messages
- Security

Device software
Billing groups

Amazon IoT > Manage > Things

Things (19) Info

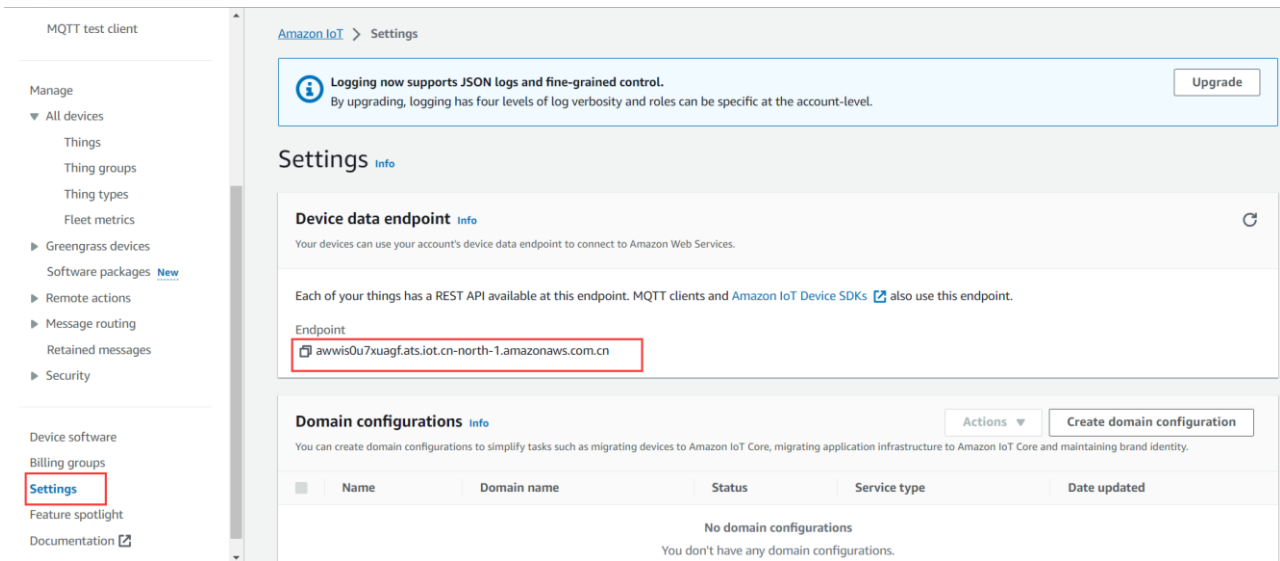
Advanced search Run aggregations Edit Delete Create things

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 Amazon IoT.

Filter things by: name, type, group, billing, or searchable attribute.

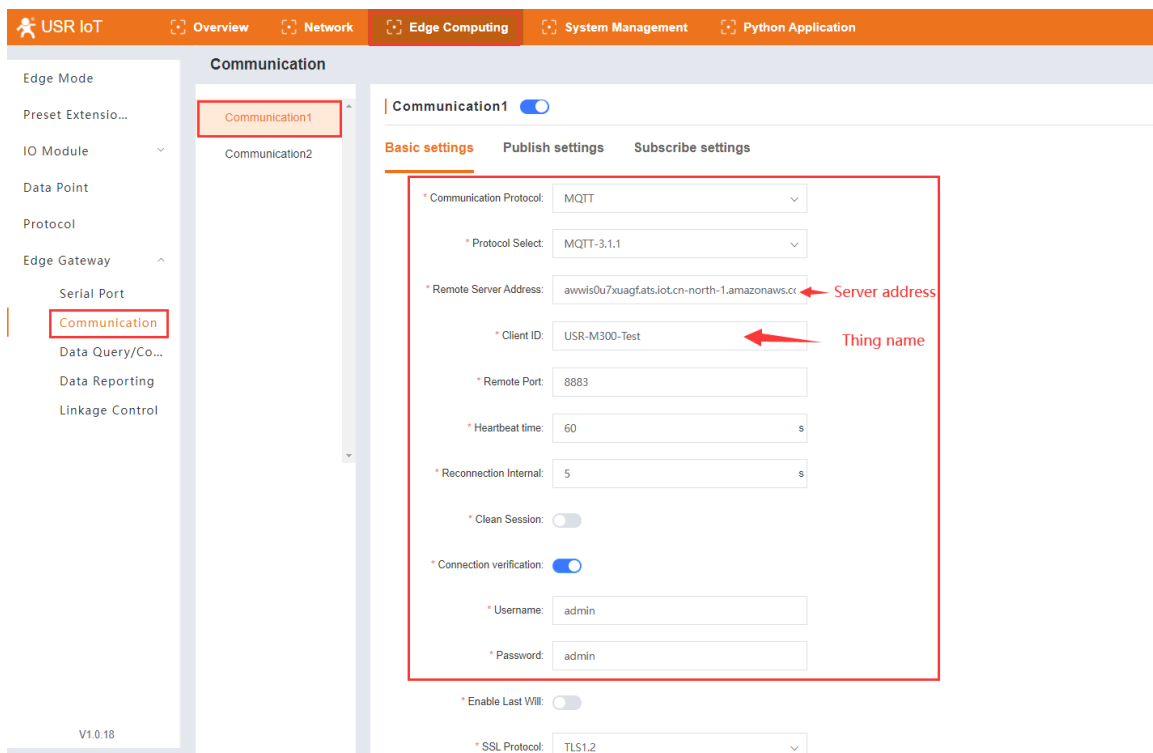
Name	Thing type
<input checked="" type="checkbox"/> USR-M300-Test	type_none
<input type="checkbox"/> USR-M300	type_none
<input type="checkbox"/> 410s_RT_H7_2	-
<input type="checkbox"/> M100-T	-
<input type="checkbox"/> USR-M100	type_none
<input type="checkbox"/> USR-M100-PUSR	-
<input type="checkbox"/> 510-sgb-test	-
<input type="checkbox"/> 540-4-test	type_none
<input type="checkbox"/> W610test	-
<input type="checkbox"/> 1064-1	-

- Find the server domain address that M300 device needs to connect in **Settings**, port defaults to **8883**.



3.3. Device Configuration

- (1) Enable MQTT communication protocol.
- (2) Configure the **Client ID**, **Remote port 8883** and **Remote server address**.
- (3) Enable **Connection verification**, **Username** and **Password** can be custom.
- (4) Add the SSL certificates, configure the **SSL protocol** to **TLS1.2**, **Self signed certificate**. Upload **AmazonRootCA1.pem** in **CA File**.



* Reconnection Interval: s

* Clean Session:

* Connection verification:

* Username:

* Password:

* Enable Last Will:

* SSL Protocol:

* Authentication Mode:

* CA File:
 ← CA1.pem

* Client Certificate File: ← certificate.pem.crt

* Client Key File: ← private.pem,key

* Report Cache Data:

(5) After above configurations, click **Apply** and then configure the topics.

(6) Publish settings: Configure the publish topic to **M300-Pub**, click **Apply**.

Communication

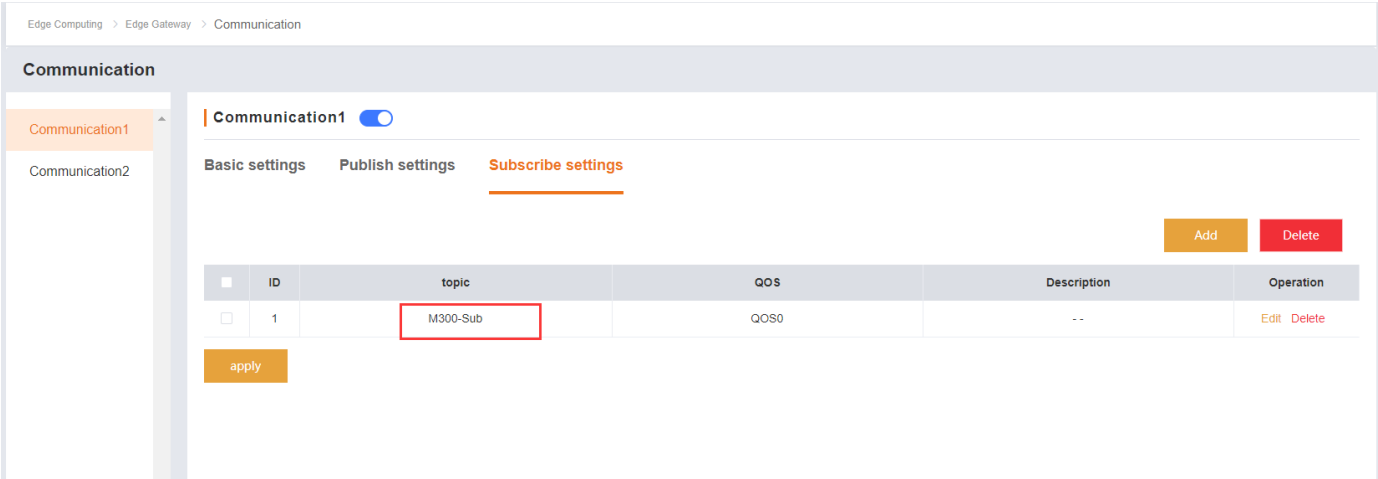
Communication1 | Communication2

Communication1

Basic settings | **Publish settings** | Subscribe settings

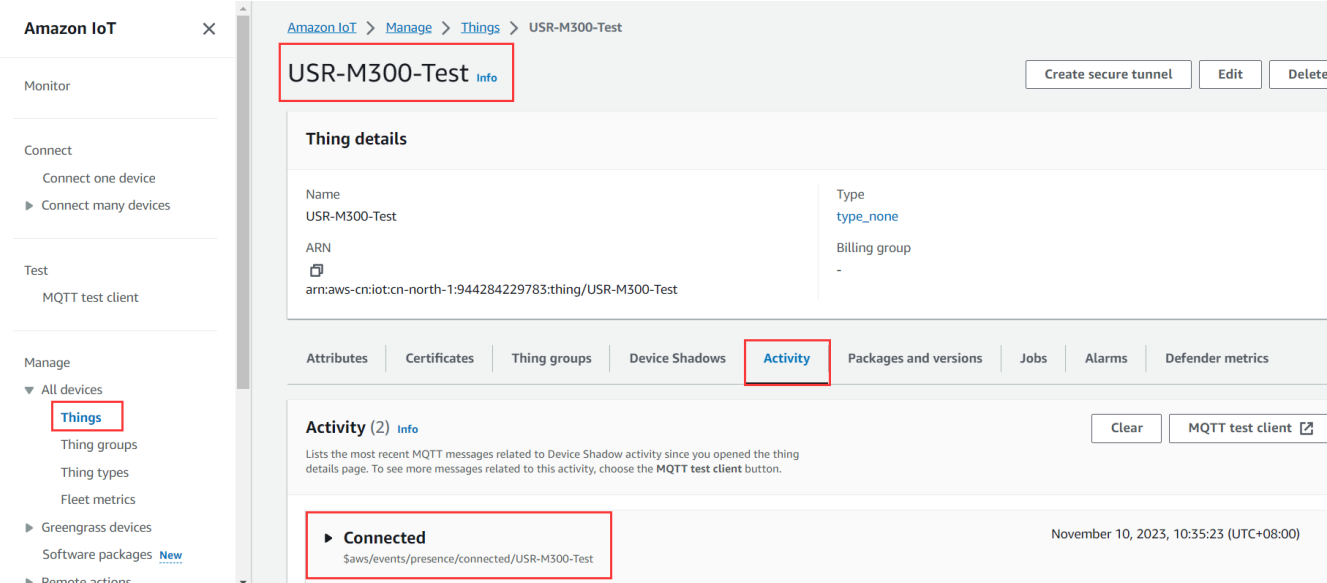
ID	topic	QOS	message retained	Description	Operation
1	M300-Pub	QOS0	not retained	..	Edit Delete

(7) Subscribe settings: Configure the subscribe topic to **M300-Sub**, click **Apply**, then restart the device to take the parameters effect.



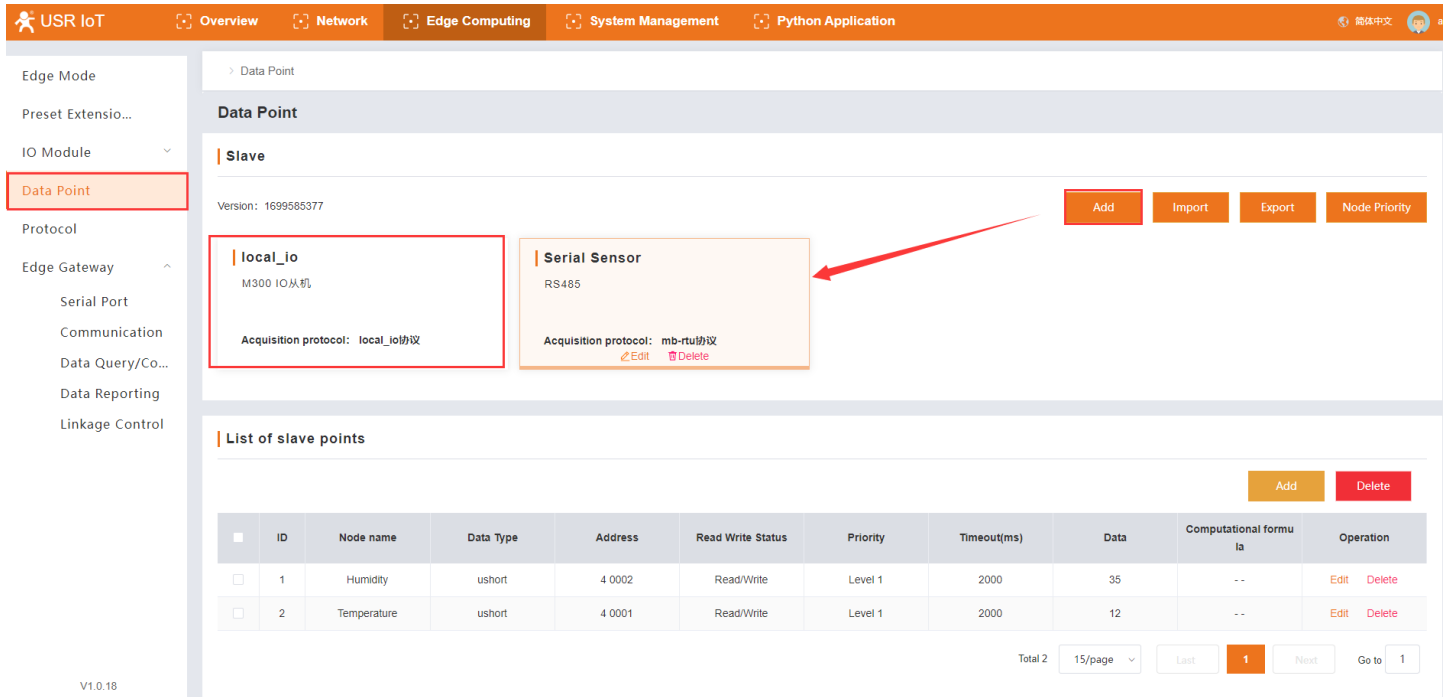
3.4. Device Activity

In **Manage->All devices->Things**, find the created device, click **Activity**, there will show **Connected**.

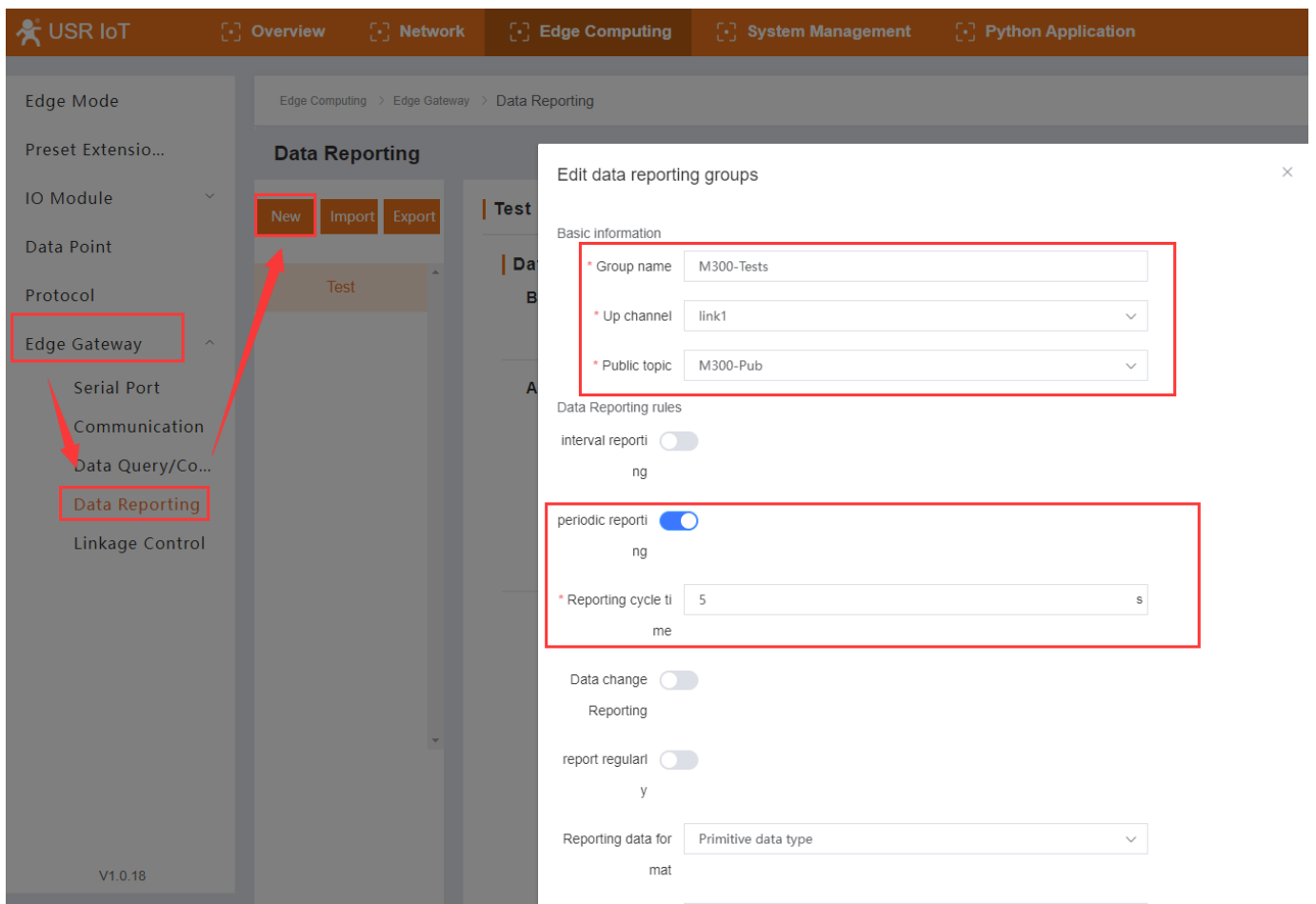


3.5. Data Transmission Test

1. Add a new serial Modbus RTU sensor in **Data Point**. And configure the Modbus registers of the sensor.



2. In Edge Gateway->Data Reporting, click **New** to add the **Data reporting groups**.



- Configure the reporting template, here we configure the device reporting the RS485 sensor data and DO1/DO1/DI1/DI2 status to AWS.

Reporting data for Primitive data type

mat

Reporting Template

```
{
  "Temperature": "Temperature",
  "Humidity": "Humidity",
  "DO01": "DO01",
  "DO02": "DO02",
  "DI01": "DI01",
  "DI02": "DI02"
}
```

cancel sure

- After configuring the reporting template, add the data points that need to be reported in **Node Table**.

```
"Humidity": "Humidity",
"DO01": "DO01",
"DO02": "DO02",
"DI01": "DI01",
"DI02": "DI02"
}
```

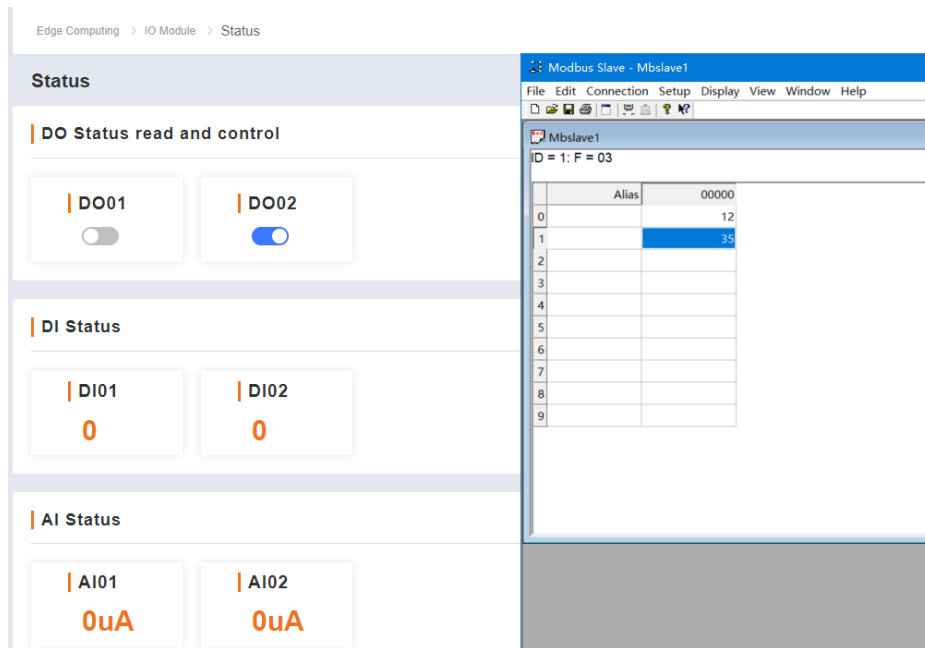
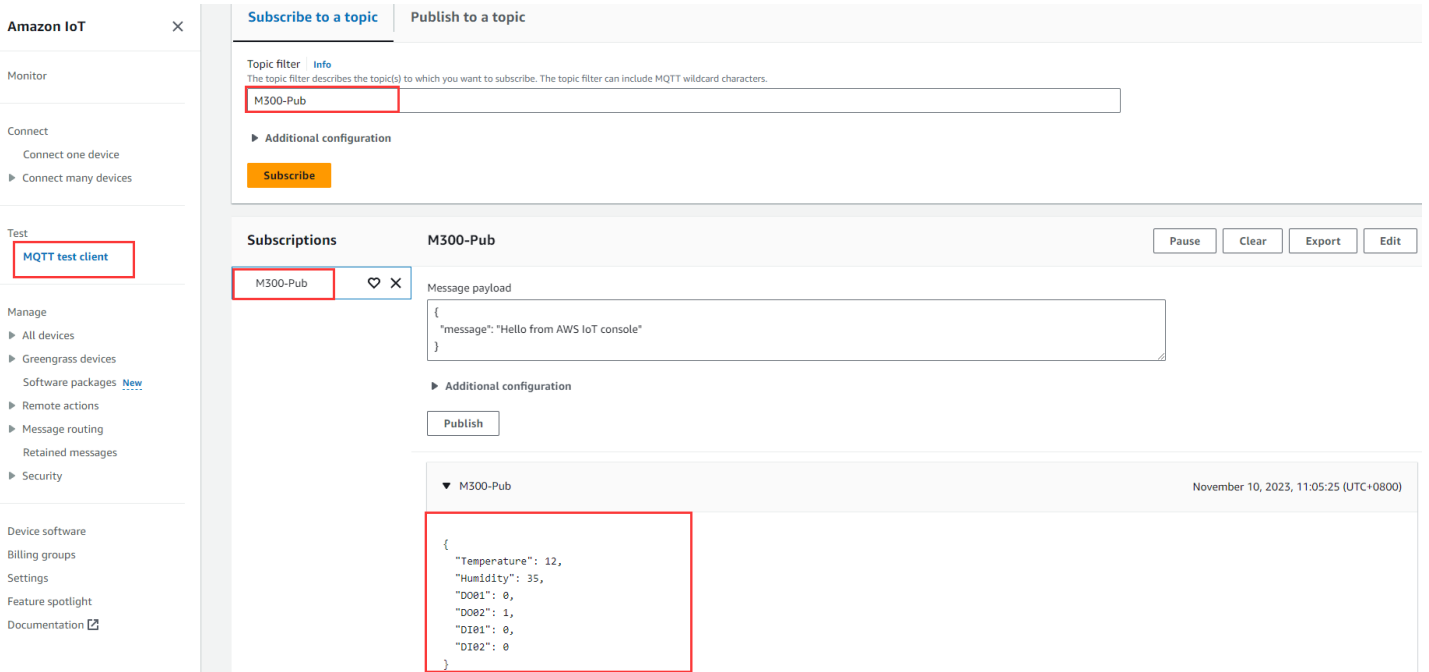
Node Table

Add Import Export Delete

ID	Node name	Slave Name	Data Type	Read Write Status	Operation
1	Humidity	Serial Sensor	ushort	Read/Write	Delete
2	Temperature	Serial Sensor	ushort	Read/Write	Delete
3	DO02	local_jo	bit	Read/Write	Delete
4	DO01	local_jo	bit	Read/Write	Delete
5	DI02	local_jo	ulong-ABCD	Only Read	Delete
6	DI01	local_jo	ulong-ABCD	Only Read	Delete
7	AI02	local_jo	float-ABCD	Only Read	Delete
8	AI01	local_jo	float-ABCD	Only Read	Delete

Total 8 10/page Last 1 Next Go to 1

- After configuring all the parameters, restart the M300 device to take the parameters effect.
- After the device connecting to the AWS again, click **MQTT test client** in **Test**, subscribe the publish topic of M300: **M300_Pub**. Then we can receive the reporting message from M300.



- M300 also supports writing the register values from the server side, we can enable **Data Query/Control** function in M300 device firstly. Configure the writing and reading JSON message according to the JSON format in below interface.

Edge Mode

Preset Extension...

IO Module

Data Point

Protocol

Edge Gateway

- Serial Port
- Communication
- Data Query/Co...**
- Data Reporting
- Linkage Control

Edge Computing > Edge Gateway > Data Query/Control

Data Query/Control

Data Query/Control Add

Select Channel	Public Topic	Subscribe Topic	Operation
Link one	M300-Pub	M300-Sub	Delete

Json: 下装:

```
{
  "rw_prot": {
    "Ver": "1.0.1",
    "dir": "down",
    "id": "12345",
    "r_data": [
      {
        "name": "node0101"
      },
      {
        "name": "node0102"
      }
    ],
    "w_data": [
      {
        "name": "node0101",
        "value": "38"
      },
      {
        "name": "node0102",
        "value": "52"
      }
    ]
  }
}
```

apply

Read and write JSON template

8. Publish writing and reading JSON message to the subscribe topic of M300 from AWS.

Amazon IoT

Monitor

Connect

- Connect one device
- Connect many devices

Test

- MQTT test client**

Manage

- All devices
- Greengrass devices
- Software packages [New](#)
- Remote actions
- Message routing
- Retained messages
- Security

Device software

Billing groups

Settings

Feature spotlight

Documentation

Subscribe to a topic

Publish to a topic

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.

Message payload

```
{
  "rw_prot": {
    "Ver": "1.0.1",
    "dir": "down",
    "id": "12345",
    "r_data": [
      {
        "name": "Temperature"
      },
      {
        "name": "Humidity"
      }
    ],
    "w_data": [
      {
        "name": "DO01",
        "value": "1"
      },
      {
        "name": "DO02",
        "value": "0"
      }
    ]
  }
}
```

Read →

Write →

Additional configuration

Publish

9. Receive the response from M300 device.

```
▼ M300-Pub November 10, 2023, 11:13:17 (UTC+0800)
```

```
{
  "rw_prot": {
    "dir": "up",
    "Ver": "1.0.1",
    "id": "12345",
    "w_data": [
      {
        "name": "DO01",
        "value": "1",
        "err": "0"
      },
      {
        "name": "DO02",
        "value": "0",
        "err": "0"
      }
    ],
    "r_data": [
      {
        "name": "Temperature",
        "value": "12",
        "err": "0"
      },
      {
        "name": "Humidity",
        "value": "35",
        "err": "0"
      }
    ]
  }
}
```



Your Trustworthy Smart IOT Partner



Official Website: www.pusr.com

Official Shop: shop.usriot.com

Technical Support: h.usriot.com

Inquiry Email: inquiry@usriot.com

Skype & WhatsApp: +86 13405313834

Click to view more: [Product Catalog](#) & [Facebook](#) & [Youtube](#)