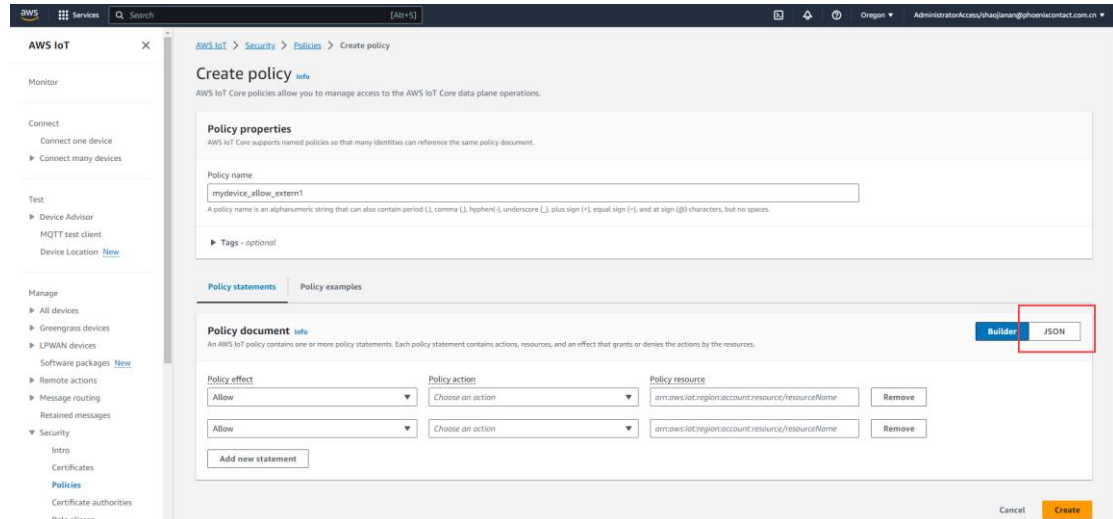


This is just an example, you must have to create a device and certificates with your AWS account to use it!

1. Log in AWS IoT, And create a goods.
 - 1) You should create a policy first.



Create policy

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:

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

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.

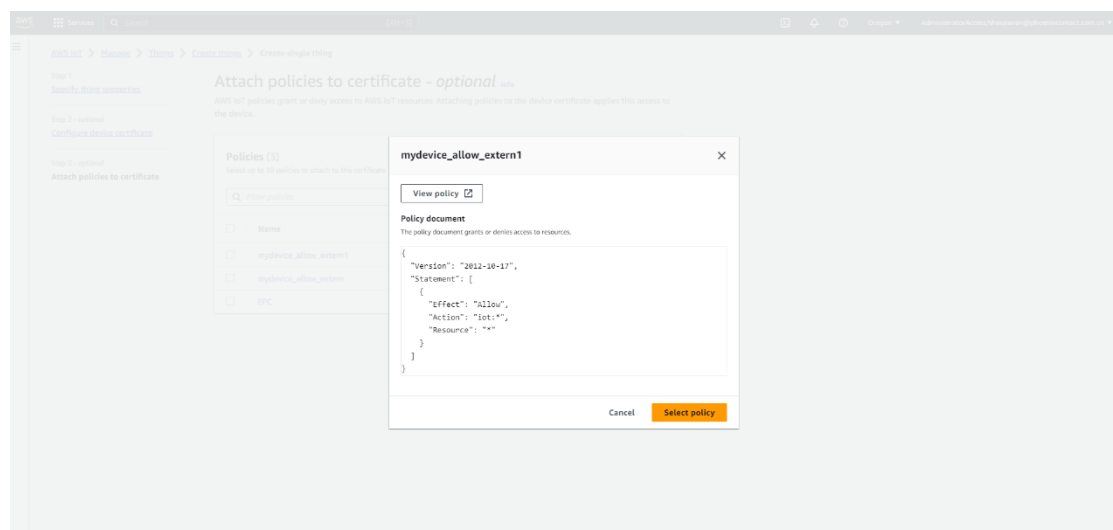
Builder | **JSON**

Policy effect	Policy action	Policy resource	
Allow	Choose an action	arn:aws:iot:region:account:resource/resourceName	Remove
Allow	Choose an action	arn:aws:iot:region:account:resource/resourceName	Remove

Add new statement

Cancel Create

- 2) The content of the policy



Attach policies to certificate - optional

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

Policies (3)

Select up to 10 policies to attach to this certificate.

mydevice_allow_extern1

View policy

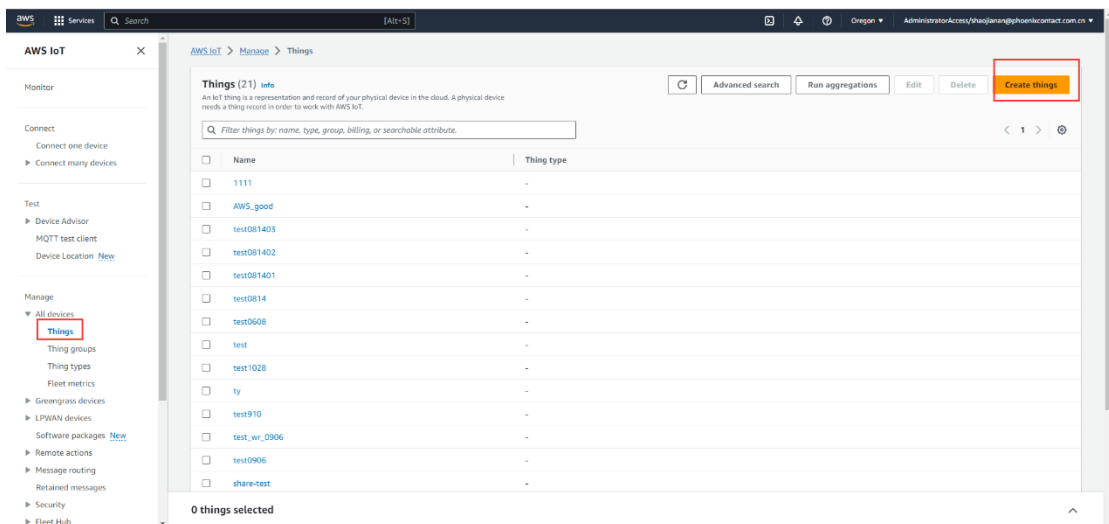
Policy document

The policy document grants or denies access to resources.

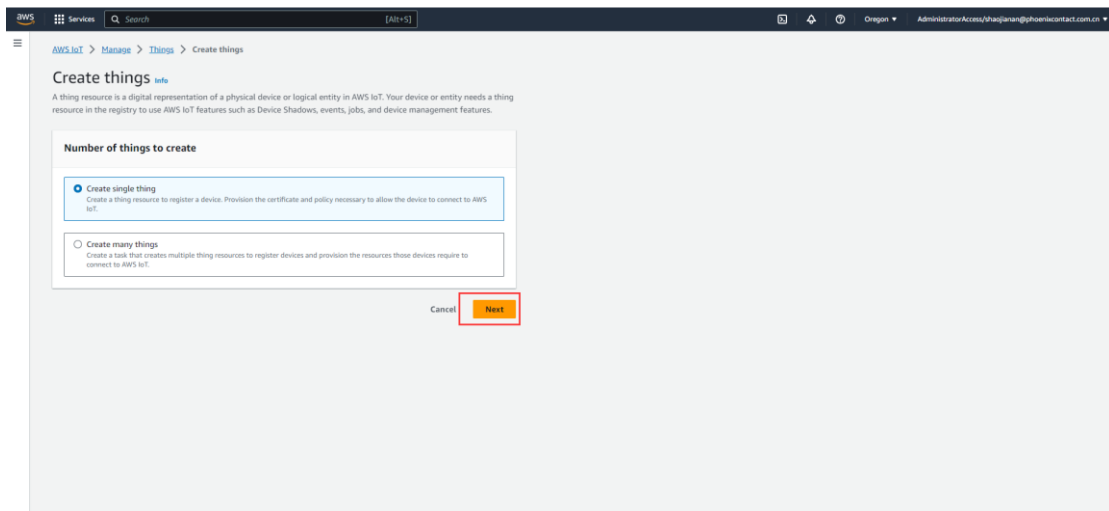
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "allow",
      "Action": "iot:*",
      "Resource": "*"
    }
  ]
}
```

Cancel Select policy

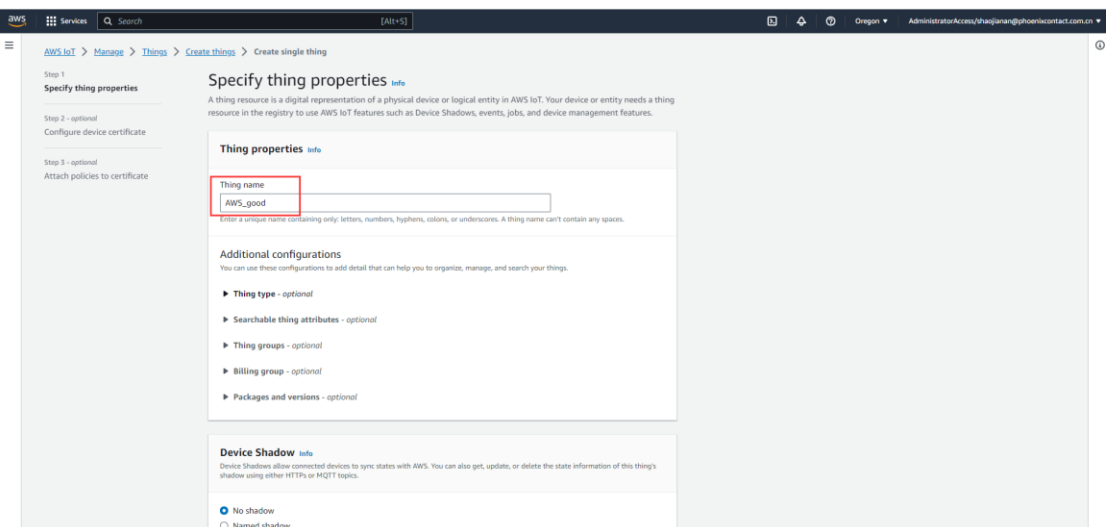
- 3) Create a new good.



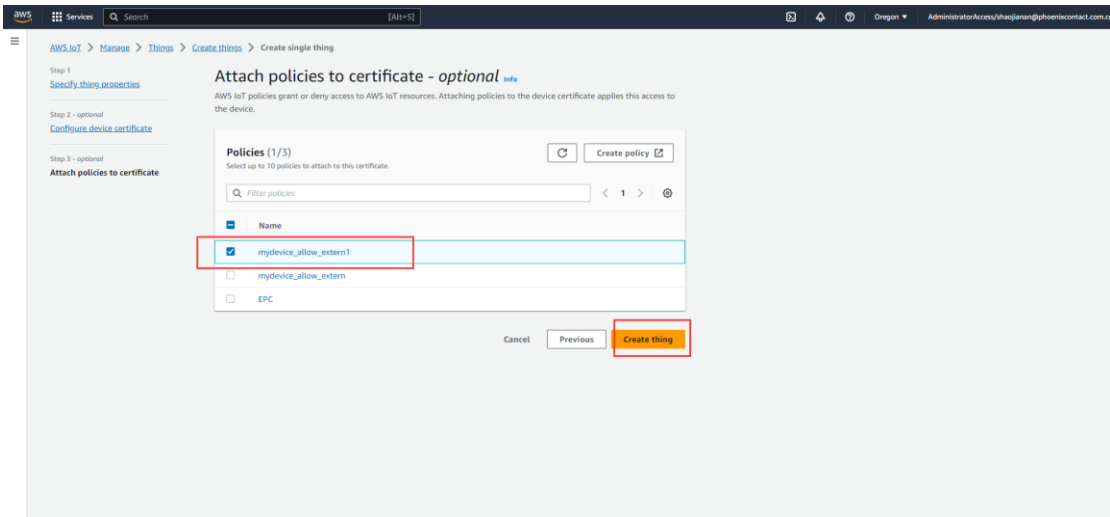
4) Choose the “Create single thing”,and then click the Next



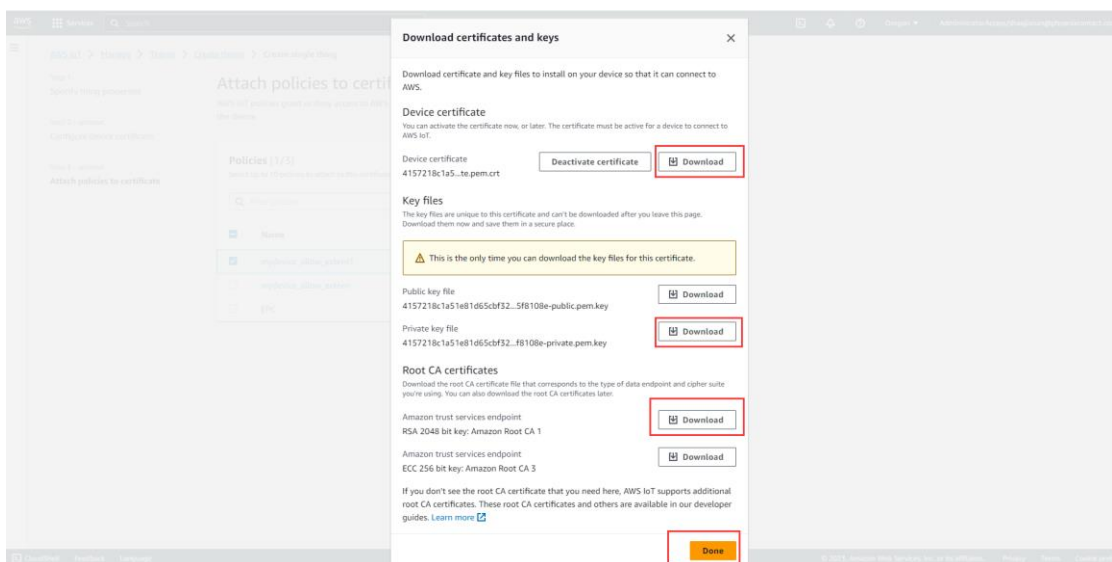
5)Then fill in the name of the goods (device_id in the example project), and then click the Next.



6)Select the policy you want to attach to the certificate,and click the “Create thing”.



7) Click Create a certificate to generate the certificate and key. And download The certificate for the item, Private key and The root CA of AWS IoT to local and rename them with a shorter name, then click activation. Attention: you should add the additional policies.



8) Open the project of PLCnext Engineer, Like IIOT_TEST_AWS_PUB_4.pcwex. open the programs variables list, you can see ca_path, cert_path and key_path. Please put these three files you downloaded before to /opt/plcnext/certs/ in PLCnext. And fill in the correct initial value in variables list, like STRING#'/opt/plcnext/certs/certificate.pem.crt'.

private.pem.key	2 KB	2023/8/14 9:54:28
certificate.pem.crt	2 KB	2023/8/14 9:54:29
AmazonRootCA1.pem	2 KB	2023/6/8 15:48:07

ca_path	STRING	本地	<input type="checkbox"/>	STRING#'/opt/plcnext/certs/AmazonRootCA1.pem'
cert_path	STRING	本地	<input type="checkbox"/>	STRING#'/opt/plcnext/certs/certificate.pem.crt'
key_path	STRING	本地	<input type="checkbox"/>	STRING#'/opt/plcnext/certs/private.pem.key'

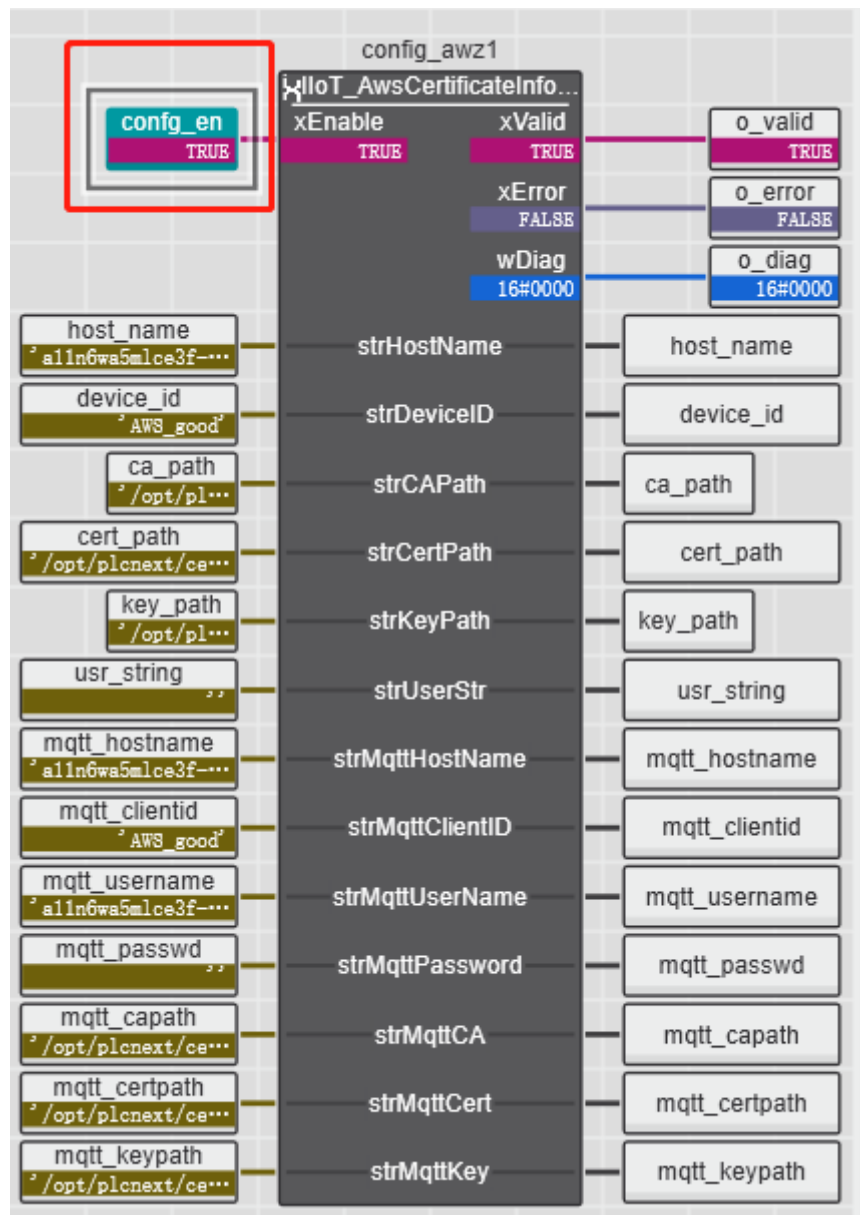
9) Replace the host_name by your own AWS account.

host_name	STRING	本地	<input type="checkbox"/>	STRING# 'a1n6wa5mlce3f-ats.iot.us-west-2.amazonaws.com'
device_id	STRING	本地	<input type="checkbox"/>	STRING# 'AWS_good'
ca_path	STRING	本地	<input type="checkbox"/>	STRING# '/opt/plcnext/certs/AmazonRootCA1.pem'
cert_path	STRING	本地	<input type="checkbox"/>	STRING# '/opt/plcnext/certs/certificate.pem.crt'
key_path	STRING	本地	<input type="checkbox"/>	STRING# '/opt/plcnext/certs/private.pem.key'

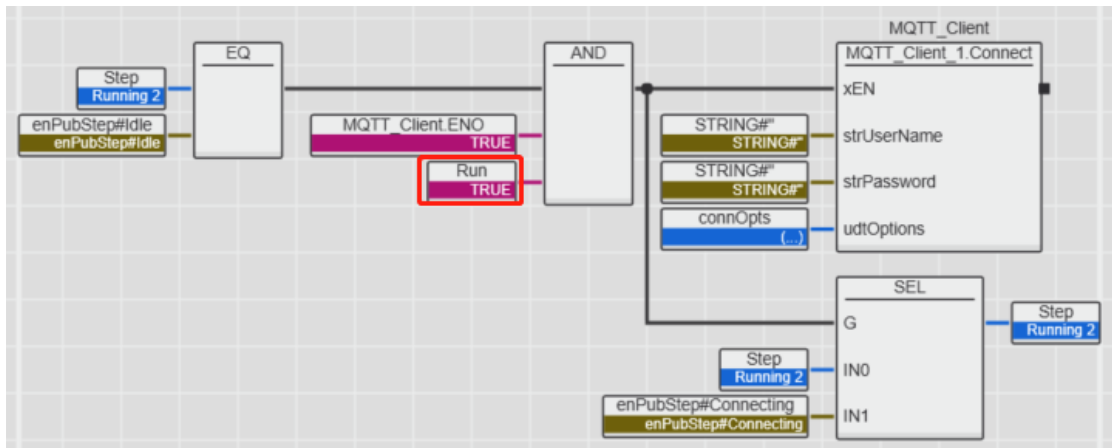
10) Connect to PLCnext and go into debug mode,

2、Connect AWS Cloud using Example Projects

- 1) Find the IIOT_AwsCertificateInfo_1 function block, set config_en as TRUE. If o_valid output TRUE, it means the certificate, key and ca is valid.



- 1) Then find the MOTT_Client_1.Connect function block, set Run as TRUE. If Step output Running 2, it means connect succeed.



- 5) Open the MQTT test client in AWS IoT. Copy the TOPIC value of project to AWS IoT. Subscribe the topic and you can receive the message from PLCnext.

TOPIC	STRING	Local	<input type="checkbox"/>	STRING#/devices/mydevice/messages/events/
-------	--------	-------	--------------------------	---

AWS IoT > MQTT test client

MQTT test client Information

You can use the MQTT test client to monitor MQTT messages delivered in your AWS account. The MQTT test client to subscribe to the MQTT message topic and publish the MQTT message to the topic.

Subscribe to the topic

Publish to the topic

Topic filter information

Topic filters describe the topics you want to subscribe to. Topic filters can include MQTT wildcards.

► Other configurations

subscribe

/devices/mydevice/messages/events/

Time out

purge

Export

edit

/devices/mydevice/messages/events/  

▼ /devices/mydevice/messages/events/

June 03, 2021, 18:38:05 (UTC+0800)

{ID:34,DT:"DT#2021-06-03-18:38:15.59"}

▼ /devices/mydevice/messages/events/

June 03, 2021, 18:37:59 (UTC+0800)

{ID:33,DT:"DT#2021-06-03-18:38:09.59"}

▼ /devices/mydevice/messages/events/

June 03, 2021, 18:37:53 (UTC+0800)

{ID:32,DT:"DT#2021-06-03-18:38:03.59"}

▼ /devices/mydevice/messages/events/

June 03, 2021, 18:37:47 (UTC+0800)

{ID:31,DT:"DT#2021-06-03-18:37:57.59"}

If you want to realize subscribe function in PLCnext, you can open IIOT_TEST_AWS_SUB_2.pcwex project. Steps are the same. And you can see the message topic, message contents and message information from msgTopic, msg, msgInfo these three variables.

AWS IoT > MQTT test client

MQTT test client

Information

You can use the MQTT test client to monitor MQTT messages delivered in your AWS account. The device publishes MQTT messages to the MQTT test client to subscribe to the MQTT message topic and publish the MQTT message to the topic.

Subscribe to the topic

Publish to the topic

The subject name

The subject name identifies the message. The message load is published to this topic with Quality of Service (QoS) 0.

Q

/plcnext_test_wr/sub

Message load

1234567

► Other configurations

publish

Name	Value	Data type	Instance
msgTopic	/plcnext_test_wr/sub	MyString	demo-aws / PLC.demo_sub
msg	[...]	MyArray	demo-aws / PLC.demo_sub
msgInfo	(...)	MQTT_UDT_ME...	demo-aws / PLC.demo_sub
udiLength	7	UDINT	demo-aws / PLC.demo_sub.msgInfo
diQos	0	DINT	demo-aws / PLC.demo_sub.msgInfo
xDuplicate	FALSE	BOOL	demo-aws / PLC.demo_sub.msgInfo
xRetained	FALSE	BOOL	demo-aws / PLC.demo_sub.msgInfo

Attention: Make sure your PC time is correct, or the certificate you download is invalid.