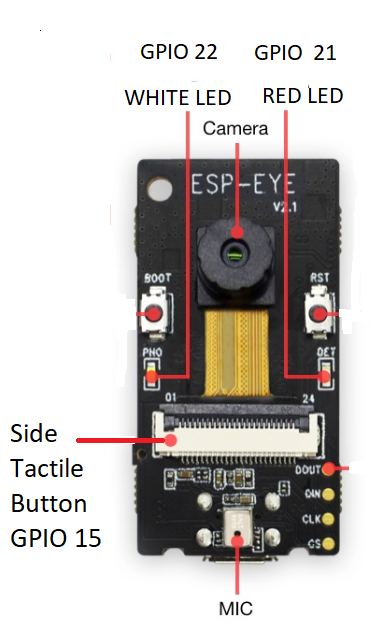


**Activity** : ESP-EYE Connection to AWS IoT Core

**Goal** : Count the number of times the button is pressed and send the count to AWS IoT Core

# LEDs and Pushbutton of ESP-EYE

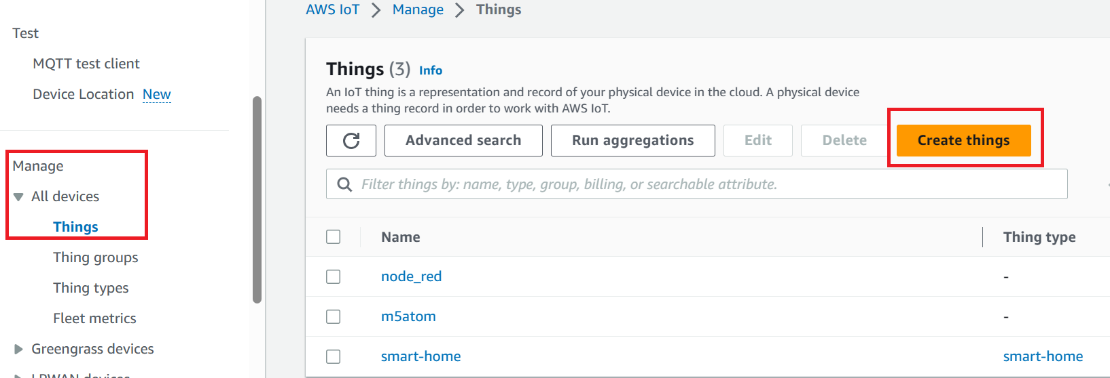


# AWS IoT Core

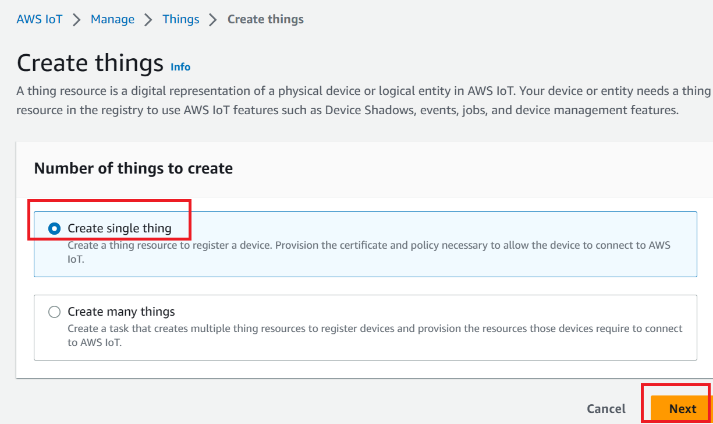
AWS IoT is an MQTT broker.

# Create a new “thing” on AWS IoT

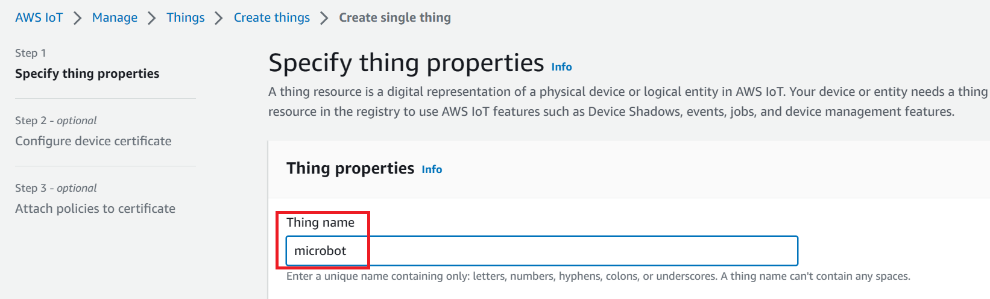
## Create a “thing”



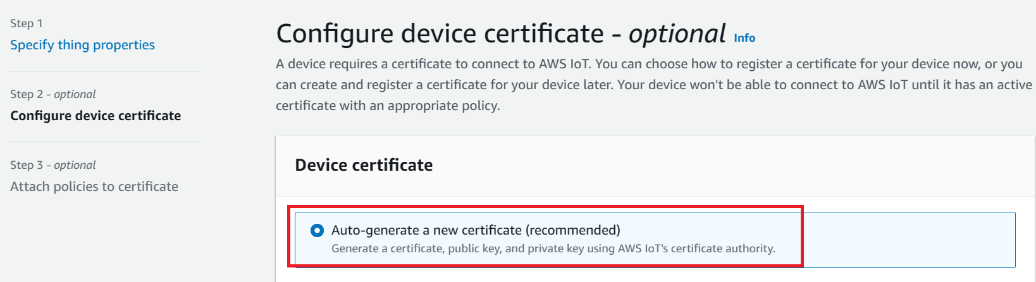
## Create a single “thing”



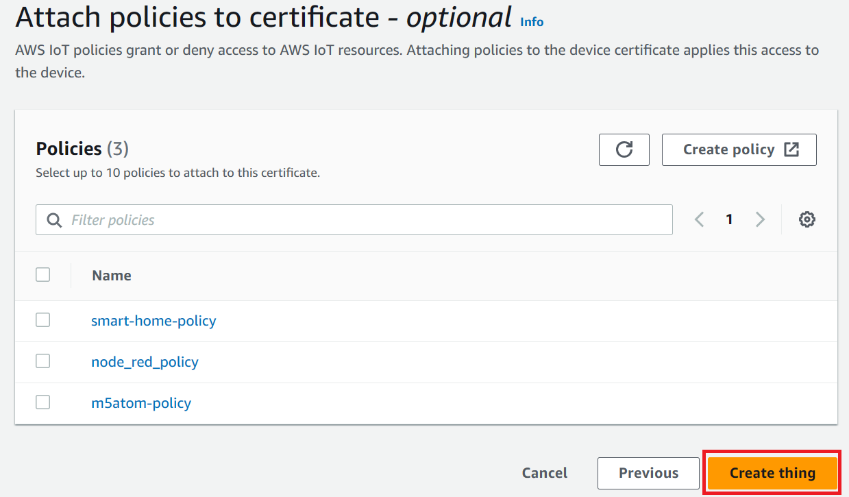
## Give the “thing” a name



## Create a certificate for the “thing”



## Create the “thing” (skip the optional creation of a new policy for the “thing”)

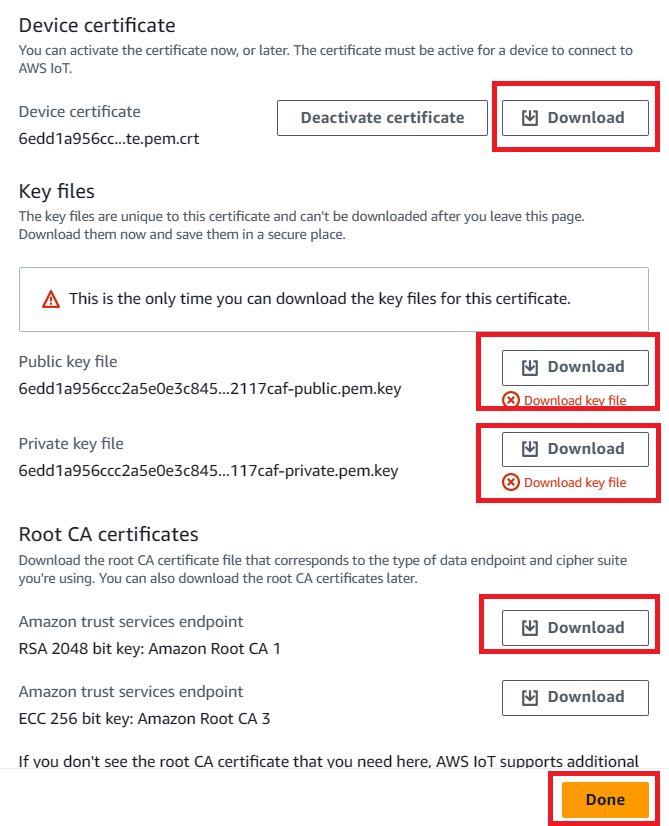


Download (1) device certificate (2) private key (3) rootCA (4) public key.

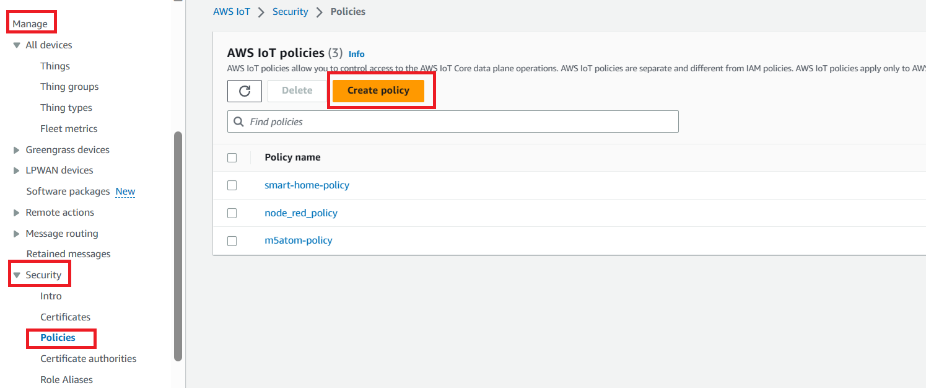
The device certificate is by default activated. If it is inactivated, you will have to activate it.

When downloading, the browser may prompt you if you really want to download and keep the file. Select “keep”.

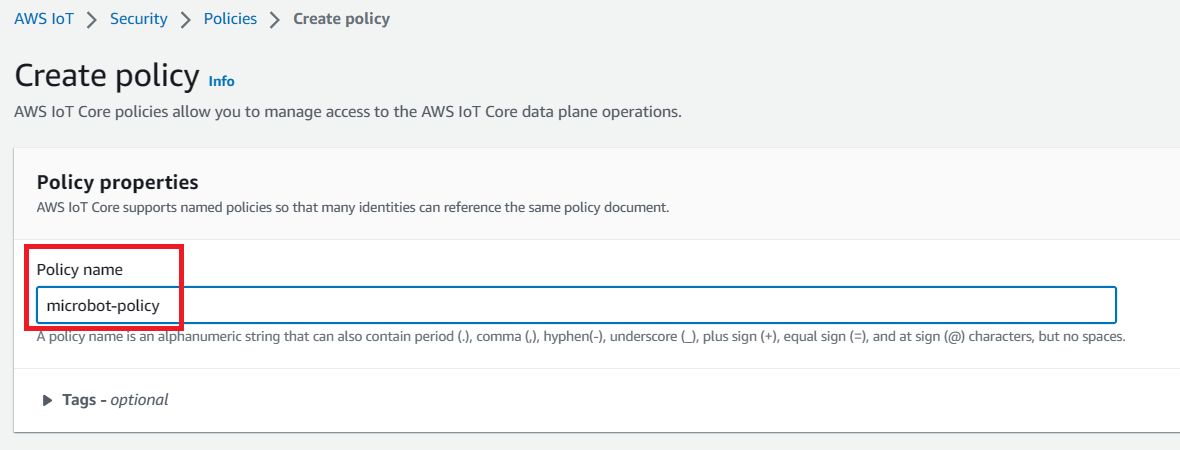
Save the files somewhere, say a “certs” folder.



# Create a Policy for the “thing”



## Give the policy a name

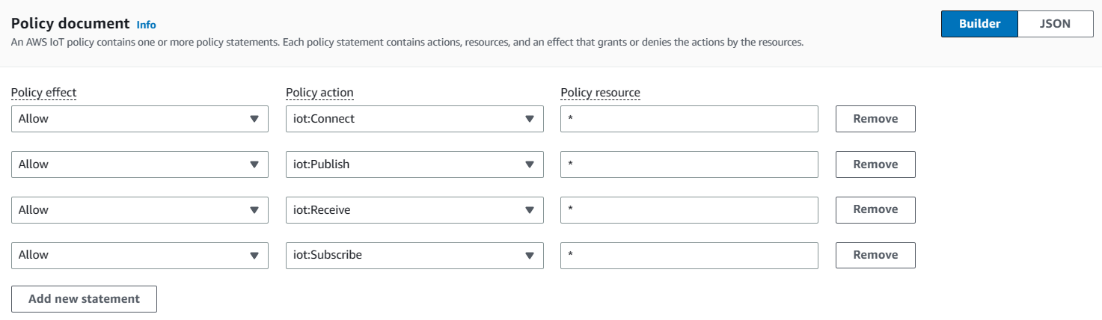


## Set Actions + Resources that are allowed

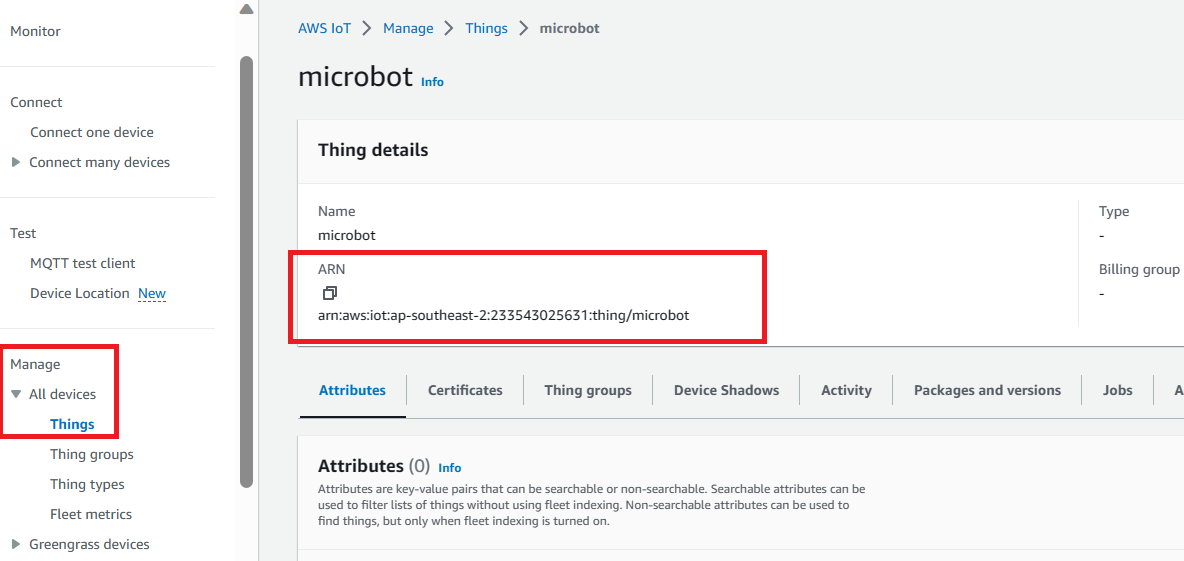
The “Policy resource” can be \*.

Add two statements:

1. Allow iot::Connect for \*
2. Allow iot::Publish for \*
3. Allow iot::Subscribe for \*
4. Allow iot::Receive for \*

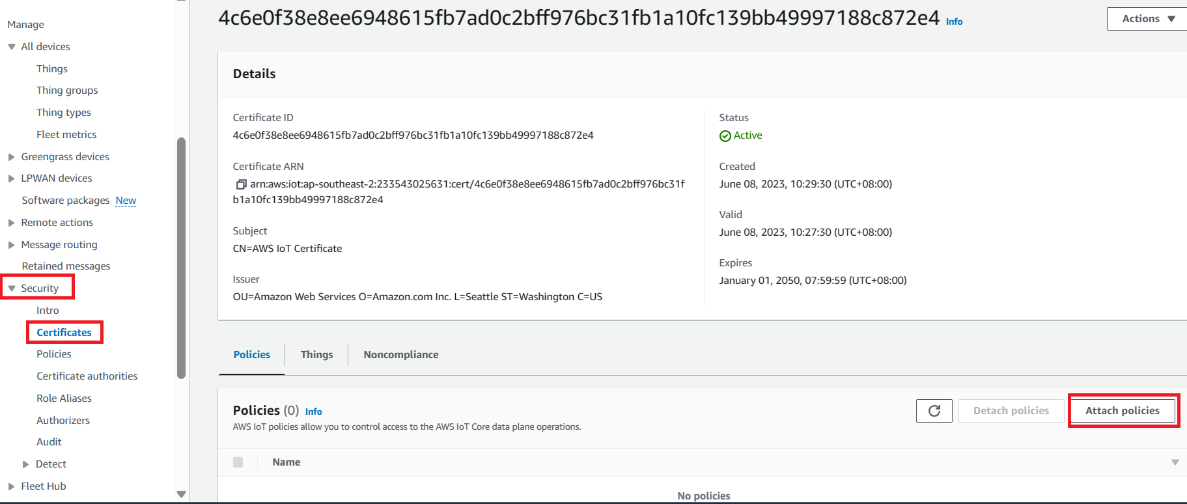


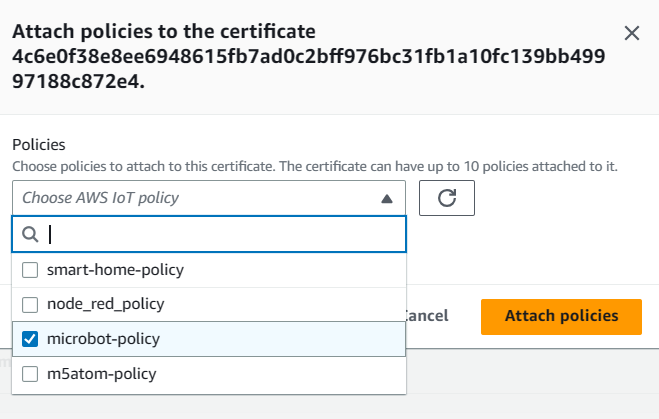
The “Policy resource” can be also be modified from the ARN of the “thing”, as shown below:



1. Allow iot::Connect, arn:aws:iot:ap-southeast-2:233543025631:client/microbot
2. Allow iot::Publishc, arn:aws:iot:ap-southeast-2:233543025631:topic/home/sensors/sensor1
3. Allow iot::Subscribe, arn:aws:iot:ap-southeast-2:233543025631:topic/home/sensors/sensor1
4. Allow iot::Receive, arn:aws:iot:ap-southeast-2:233543025631:topic/home/sensors/sensor1

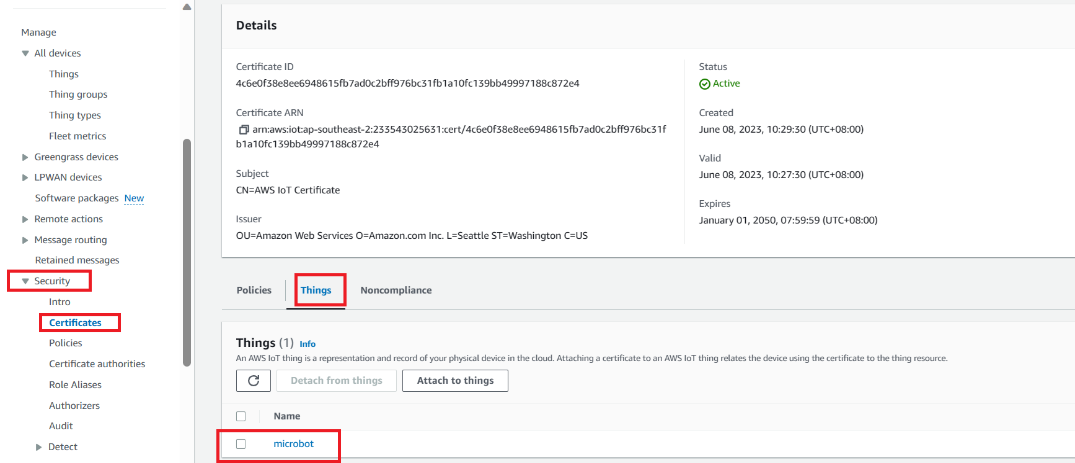
## Attach the policy to the certificate





## Attach the “thing” to the certificate

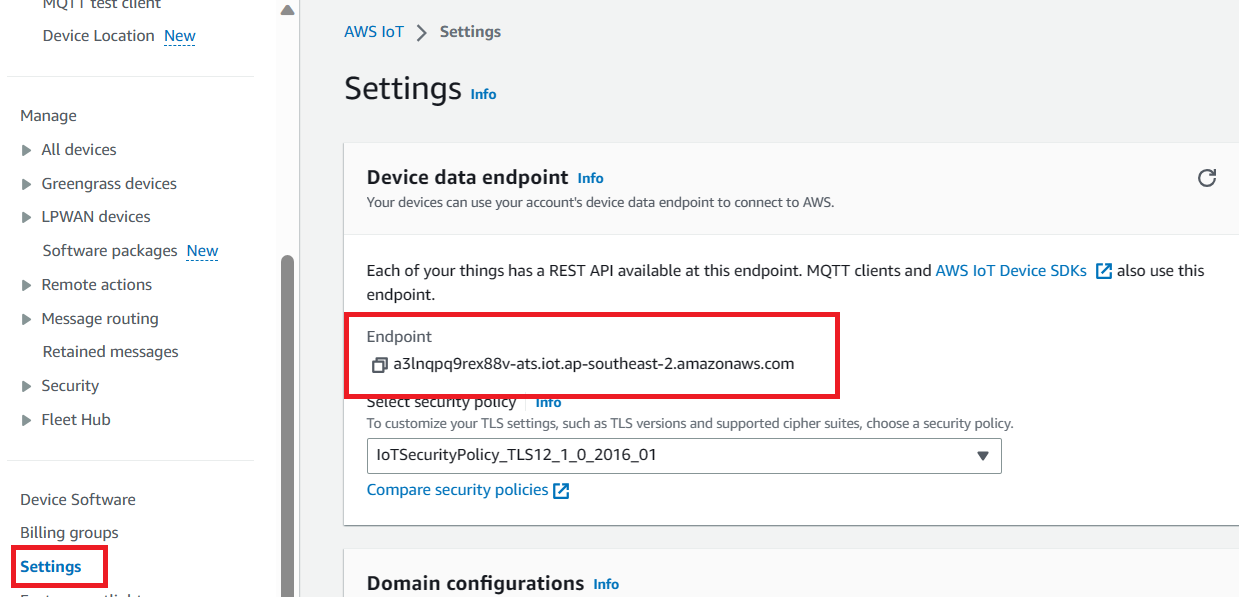
This was the default setting when we created the certificate as we created the thing. Just to be sure, check the following:



# Arduino Code



The AWS MQTT server address is obtained from the Setting Page of the AWS IoT management console, as shown below.



## Setup



## Loop



## Interrupt Service Routine



## Subscription Callback



# Include Certificates in aws\_iot\_certificates.c

In C:\Users\SCSC\Documents\Arduino\libraries\AWS\_IOT\src\ aws\_iot\_certficates.c , modify the content with the following data:

1. root-CA.pem
2. device certificate (certificate.pem.crt)
3. Private key (private.pem.key)

Between “”, put \n\ in every single line, and the final line with \n.

# Run & Test

On AWS IoT management console, go to Test section. Subscribe to topic “espeye/count”. When ESP-EYE is run, the data should be displayed there.

