



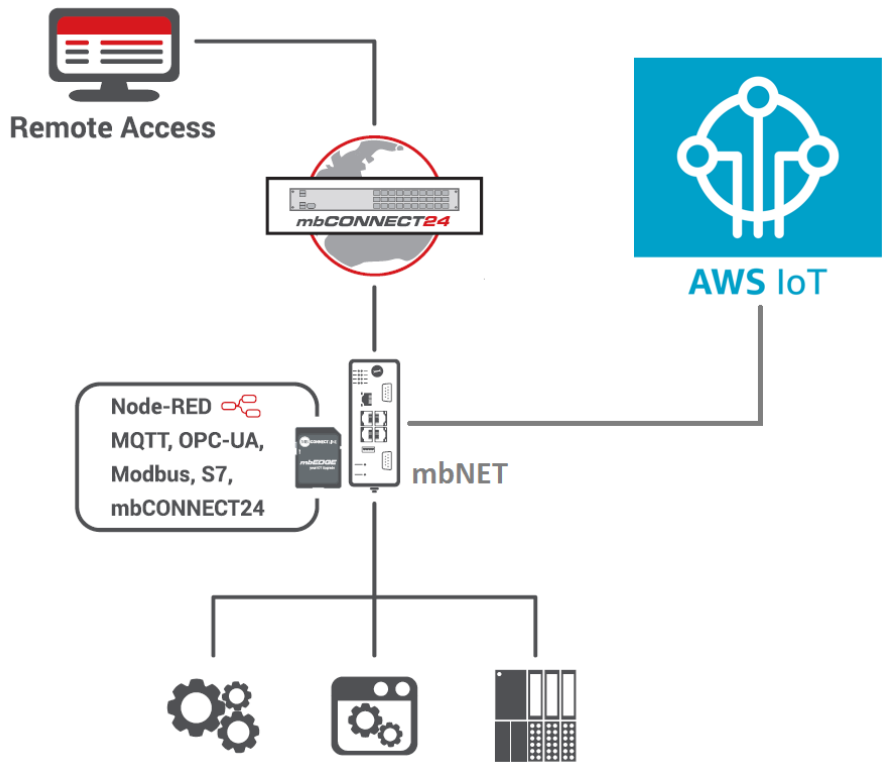
AWS IoT (EN)

(Document Rev. 8, 04.06.2020)

Generated on 04.06.2020 by Siegfried Müller

Table of Contents

1	Prerequisite	5
2	1. Create your AWS Account	6
3	2. Create AWS IoT object.....	7
3.1	2.1. Create the object by "Register a thing"	7
3.2	2.2. Create a single thing.....	8
3.3	2.3. Define your thing name. Thing Names should be unique per device. Use mbNET in this example. .8	8
3.4	2.4. Create your certificates.....	9
3.5	2.5. Download your certificates (1) and the root CA (2). After then activate (3) and your all set (4).....	10
3.6	2.6. Create your policy	12
4	3. Set MQTT Node in Node-RED.....	18
4.1	3.1 AWS endpoint settings.....	18
4.2	3.2. MQTT Client-ID settings	19
4.3	3.3. Import certificates.....	20
4.4	3.4. check connection	20
5	4. Advanced: Create and use AWS Shadow.....	22
5.1	4.1. Create our shadow document	22
5.2	4.2. Edit policy	23
5.3	4.3 Add the Node-RED Flow	25



A standard MQTT client from Node-RED is used to establish the connection between mbEDGE (node-red) and AWS. The "AWS IoT" service is used on the AWS website. Here are the rough steps for implementation:

1. AWS account for the AWS IoT service
2. Create AWS IoT object
 - a. Create certificates (for establishing a connection for authentication)
 - b. Create a guideline (which device may use which functions such as publish or subscribe)
3. Create MQTT client in Node-RED
 - a. Enter AWS endpoint as MQTT server
 - b. Enter the MQTT client ID
 - c. Import certificates
4. Advanced: Create and use AWS Shadow

First of all we need a mbNET router and the mbEDGE option SD Card. Every mbNET router starting from Hardwareversion 3 is suitable.

mbNET router RKH210

mbEDGE



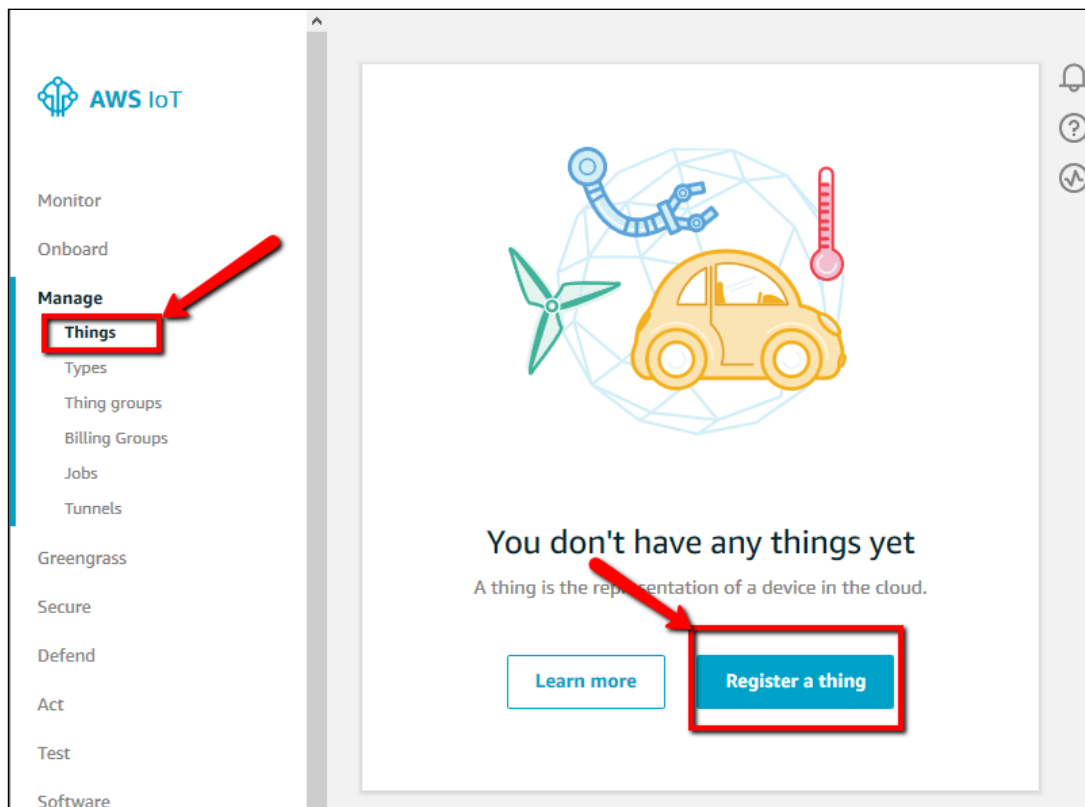
1 Prerequisite

You can either use the mbNET router as an classic router without using with the device management cloud mbCONNECT24 or with mbCONNECT24. In both ways your mbEDGE card must be enabled and the Node-RED flow editor must be accessible.

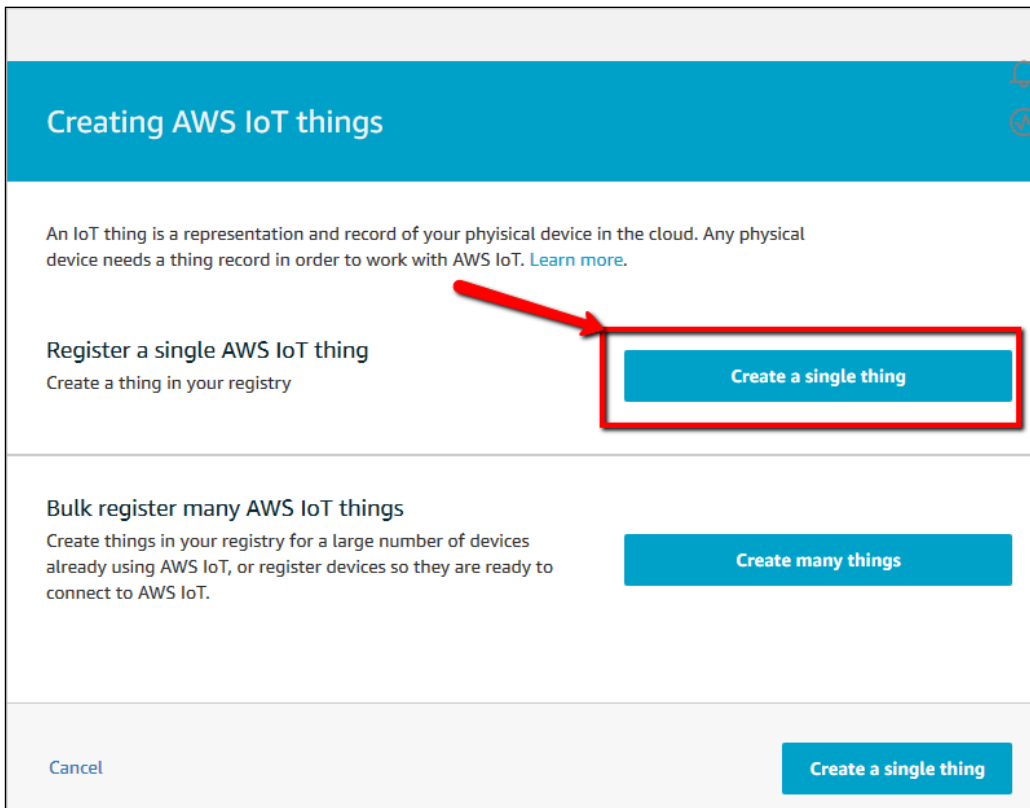
2 1. Create your AWS Account

3 2. Create AWS IoT object

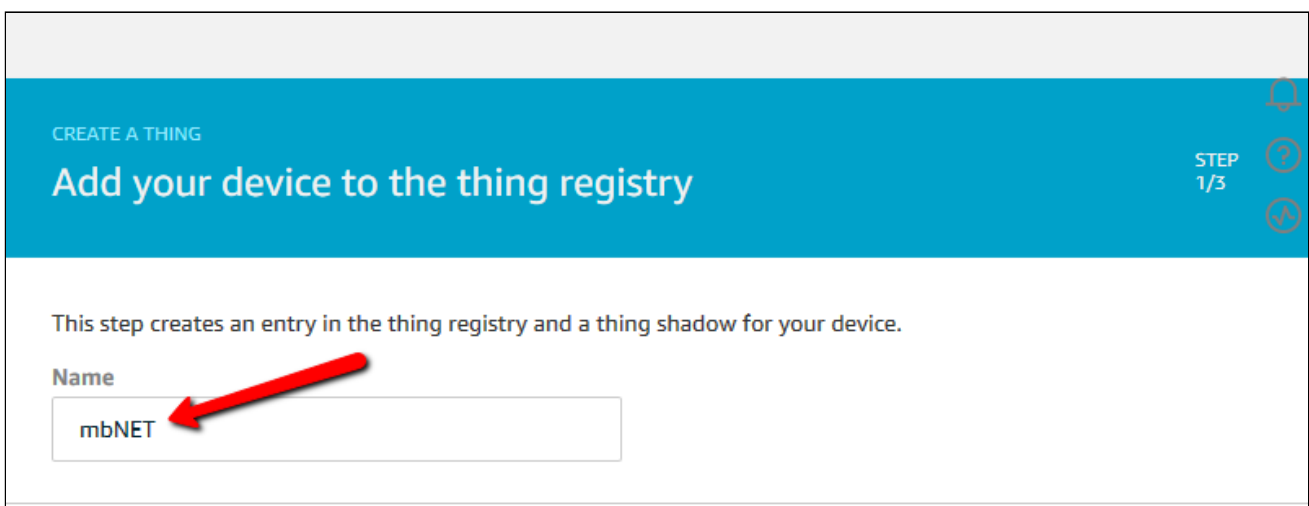
3.1 2.1. Create the object by "Register a thing"



3.2 2.2. Create a single thing



3.3 2.3. Define your thing name. Thing Names should be unique per device. Use mbNET in this example.

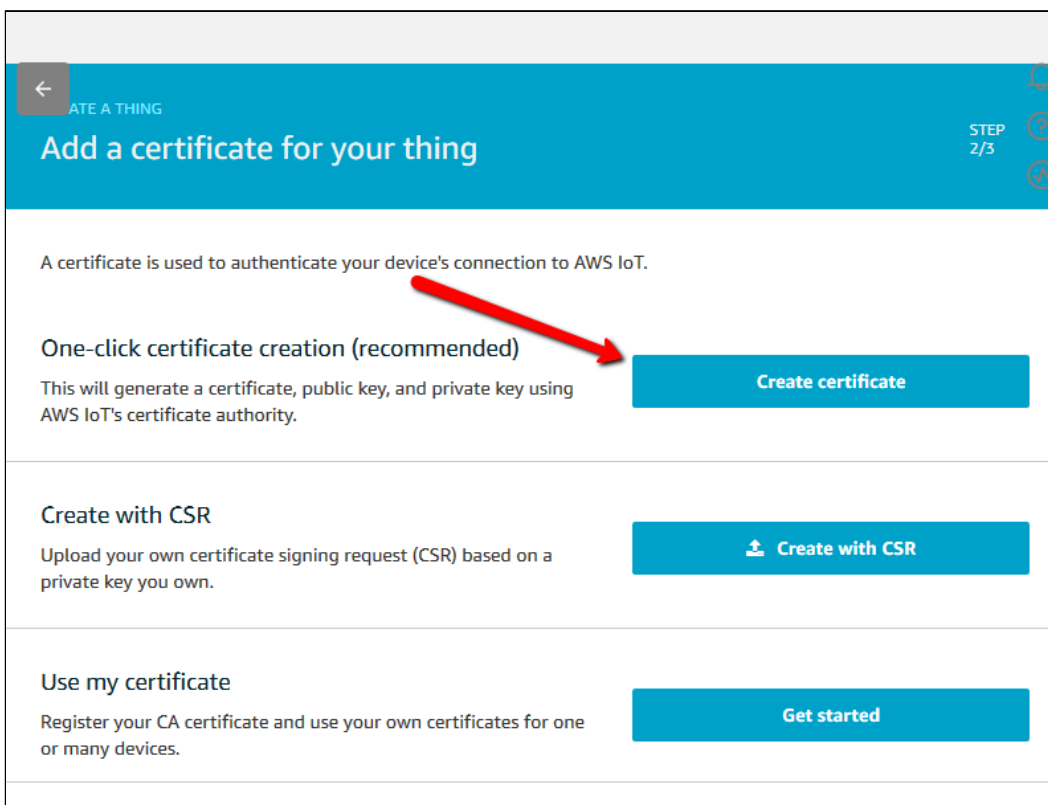


scroll down to "next" and leave all other empty or nothing.



⚠ Important
Thing Names should be unique per device. The Thing Name should also correspond with the Client ID, which must be unique.

3.4 2.4. Create your certificates



3.5 2.5. Download your certificates (1) and the root CA (2). After then activate (3) and your all set (4).

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:

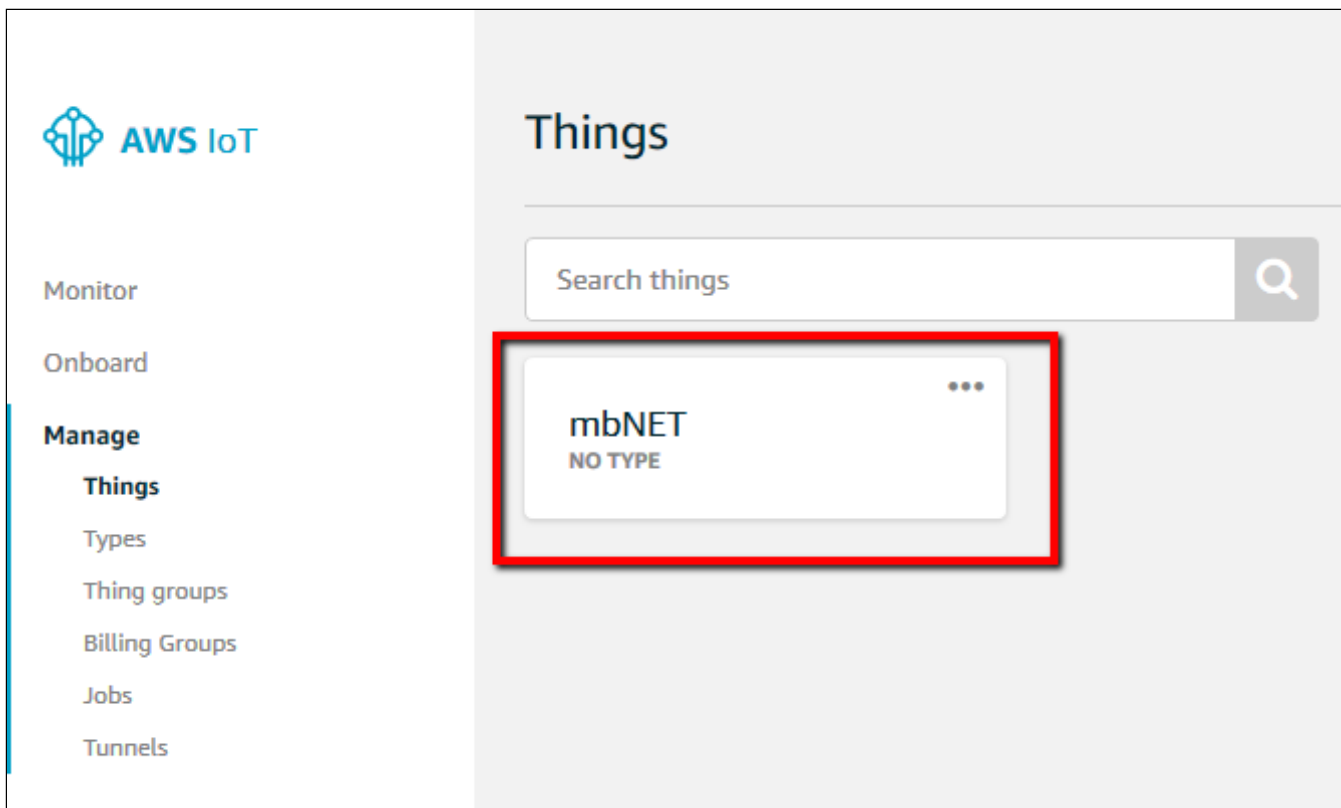
A certificate for this thing	d0377249f5.cert.pem	Download
A public key	d0377249f5.public.key	Download
A private key	d0377249f5.private.key	Download

You also need to download a root CA for AWS IoT:
A root CA for AWS IoT [Download](#)

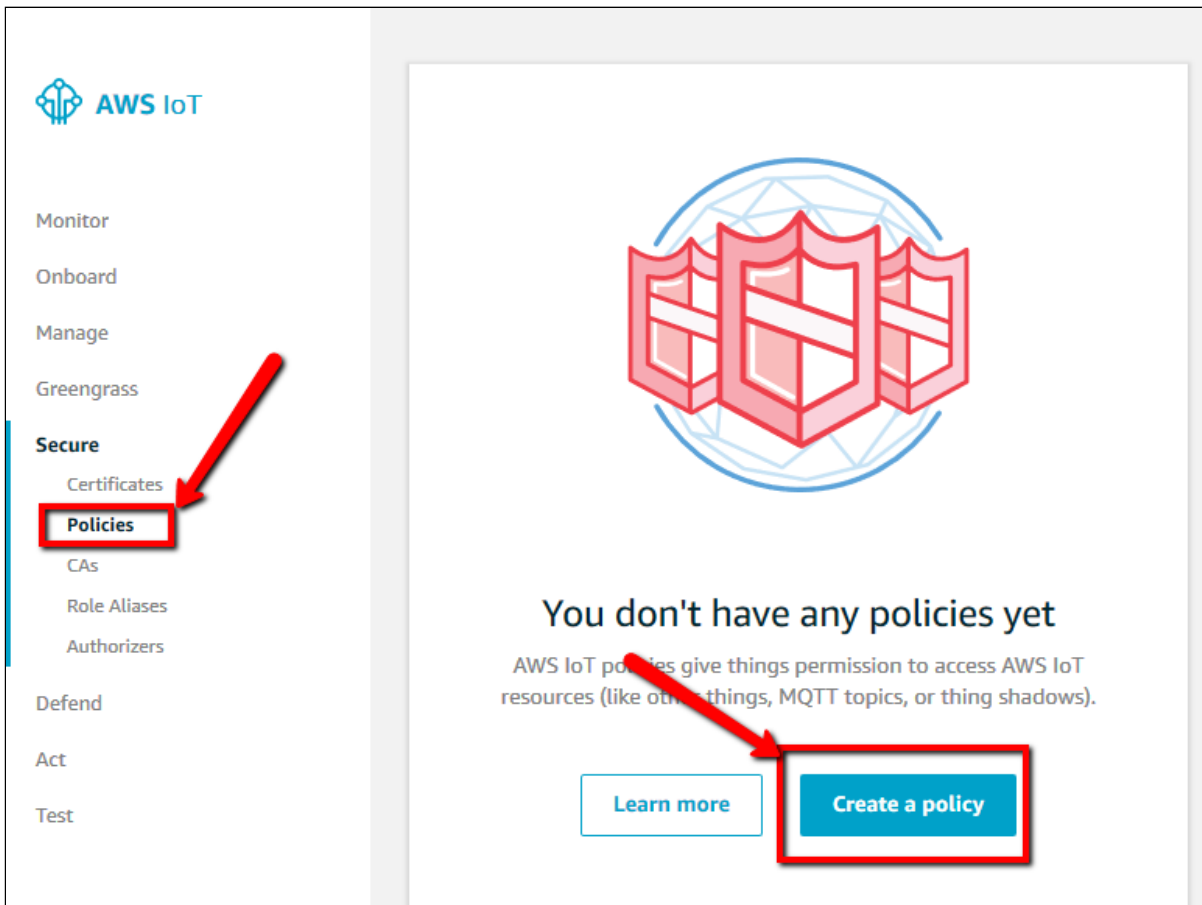
[Activate](#)

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

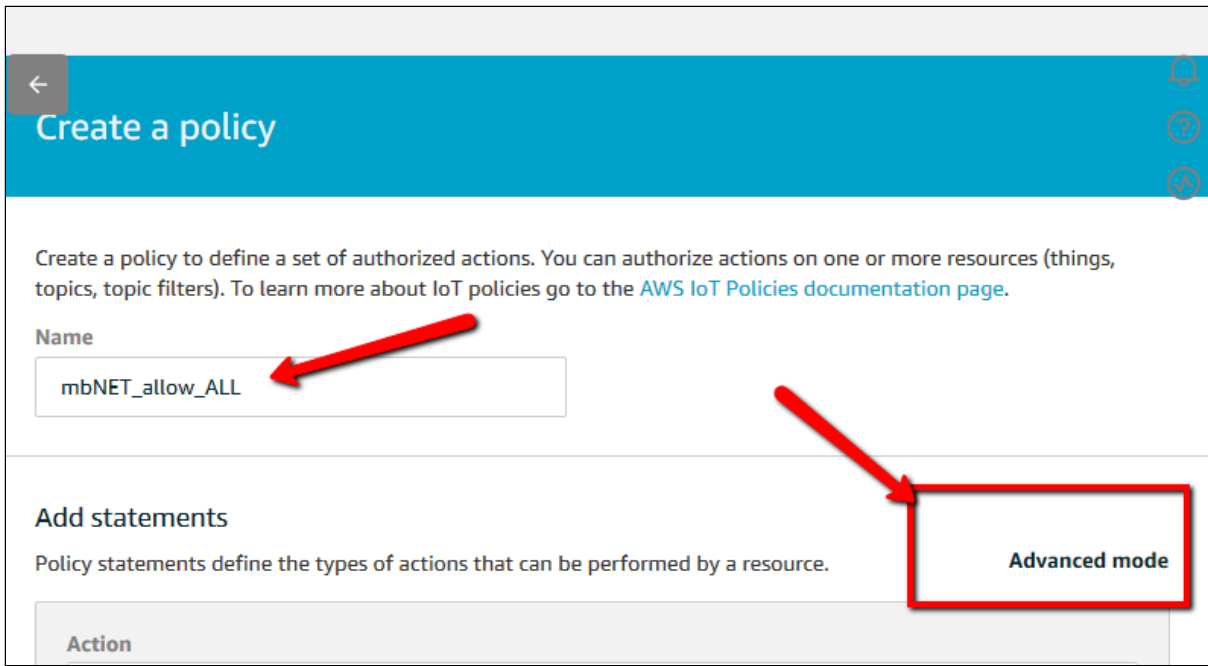
If you are done, you should see the thing "mbNET" like this.



3.6 2.6. Create your policy



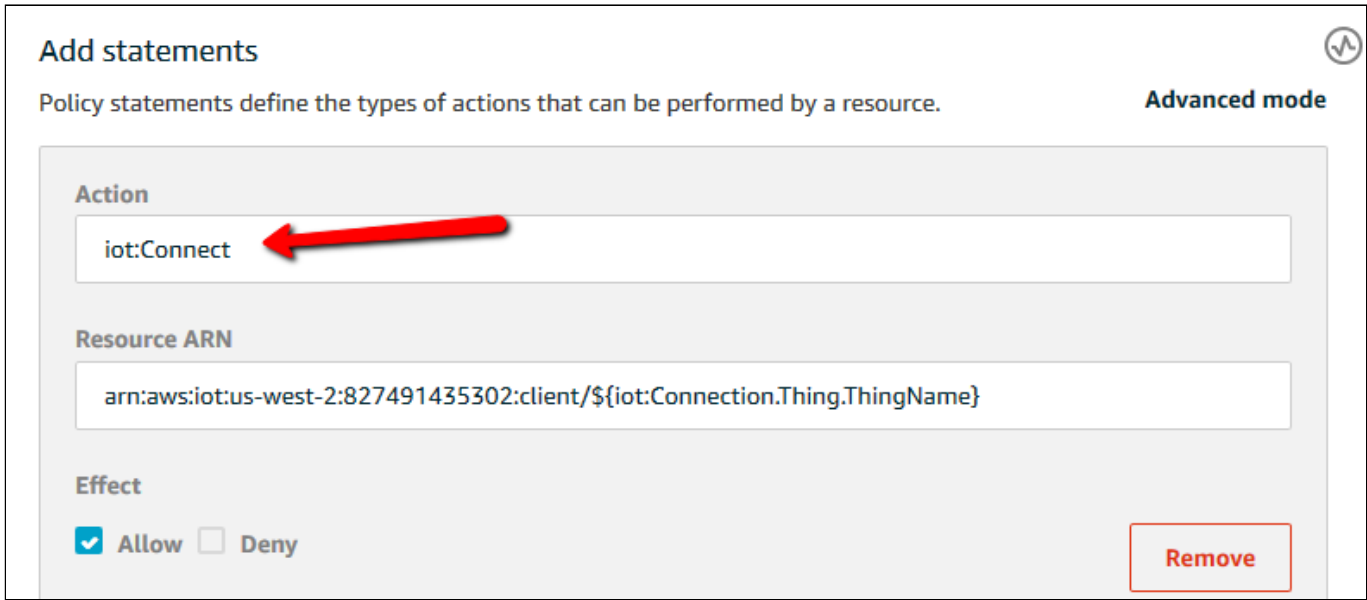
Define a name and add your policy statements



We want to differ between "Connect", "Publish/Receive" and "Subscribe". The policy handles the access to your things and topics.

In our Example we allow all to connect and access to topic "mymbNETTopic". If you want to allow all, the use the wildcard "*" instead of the topic name.

Add "iot:Connect" to the "Action" field and "Add statement". Replace the word "replaceWithAClientId" in the field "Resource ARN" with "\${iot:Connection.Thing.ThingName}"



Add "iot:Publish, iot:Receive" to the "Action" field and "Add statement". Replace the word "replaceWithATopic" in the field "Resource ARN" with "mymbNETTopic"

Action
iot:Publish, iot:Receive

Resource ARN
arn:aws:iot:us-west-2:827491435302:topic/mymbNETTopic

Effect
 Allow Deny

Remove

Warning

If you want to access all topics, replace "replaceWithATopic" with "*". But use this only for testing purposes, NOT for production use.

Add "iot:Subscribe" to the "Action" field and "Add statement". Replace the word "replaceWithATopicFilter" in the field "Resource ARN" with "mymbNETTopic"

Action
iot:Subscribe

Resource ARN
arn:aws:iot:us-west-2:827491435302:topicfilter/mymbNETTopic

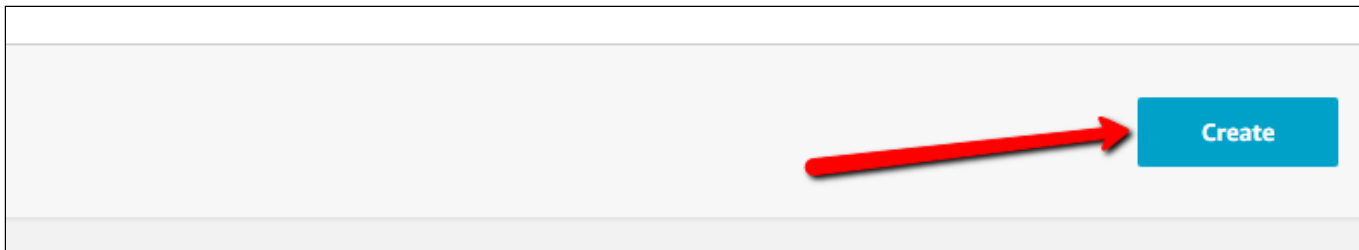
Effect
 Allow Deny

Remove

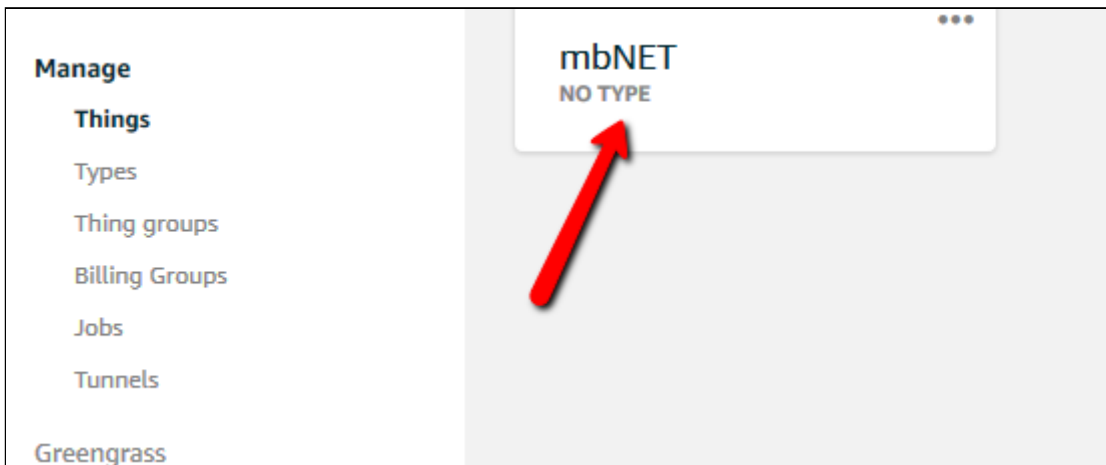
Warning

If you want to access all topics, replace "replaceWithATopic" with "*". But use this only for testing purposes, NOT for production use.

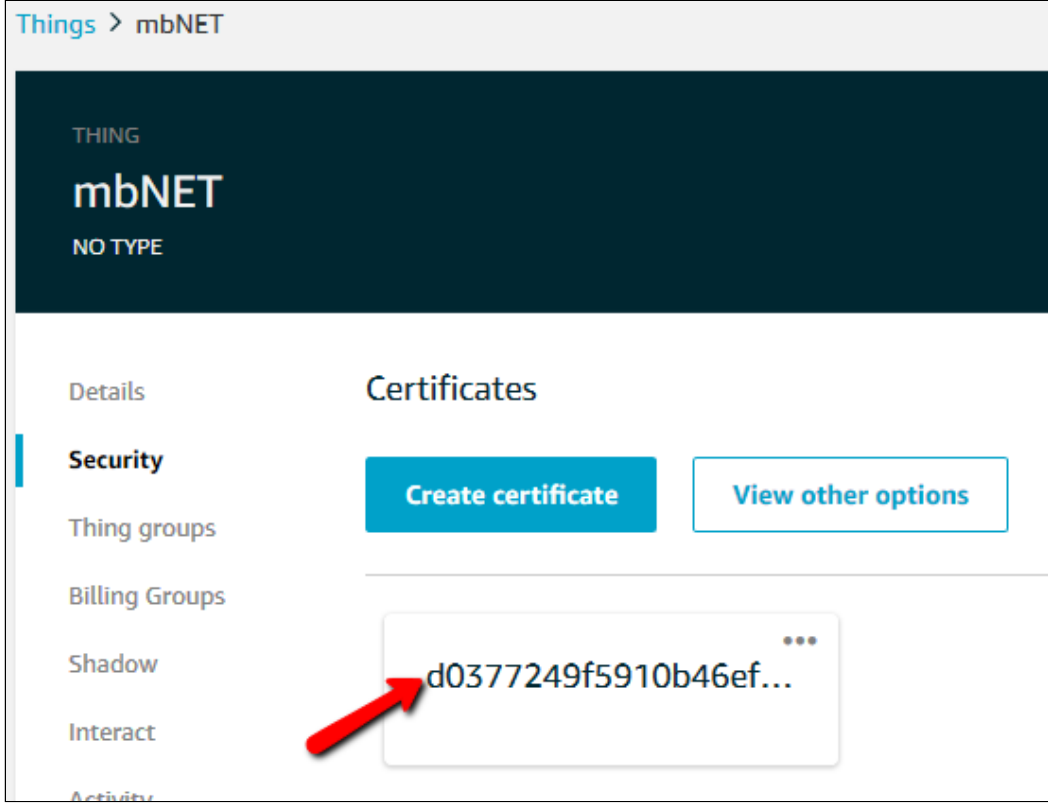
When you are finished, click on "Create" at the bottom of the website.



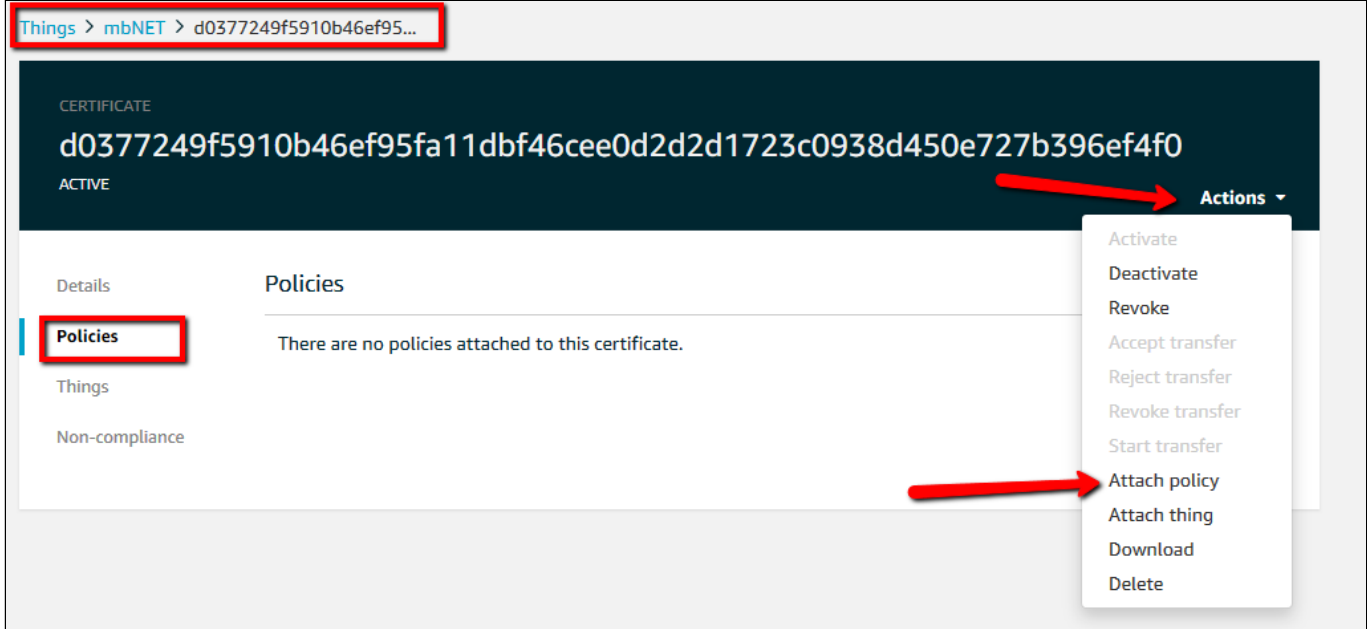
Select your thing



Select your certificate



Attach your policy to your certificate



Attach policies to certificate(s)

Policies will be attached to the following certificate(s):

d0377249f5910b46ef95fa11dbf46cee0d2d2d1723c0938d450e727b396ef4f0

Choose one or more policies

<input checked="" type="checkbox"/> mbNET_allow_ALL	View
---	----------------------

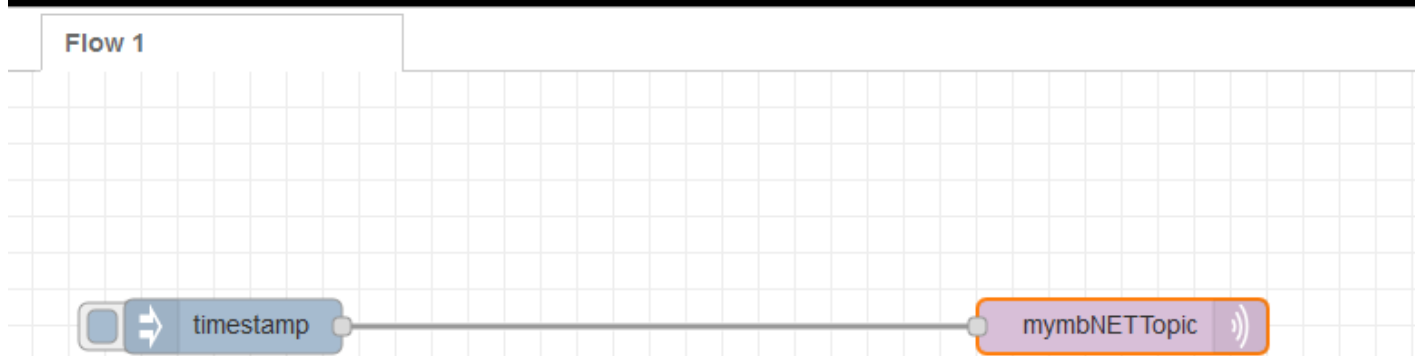


1 policy selected

Cancel

Attach

4 3. Set MQTT Node in Node-RED



Edit mqtt out node

Properties

Server

Topic

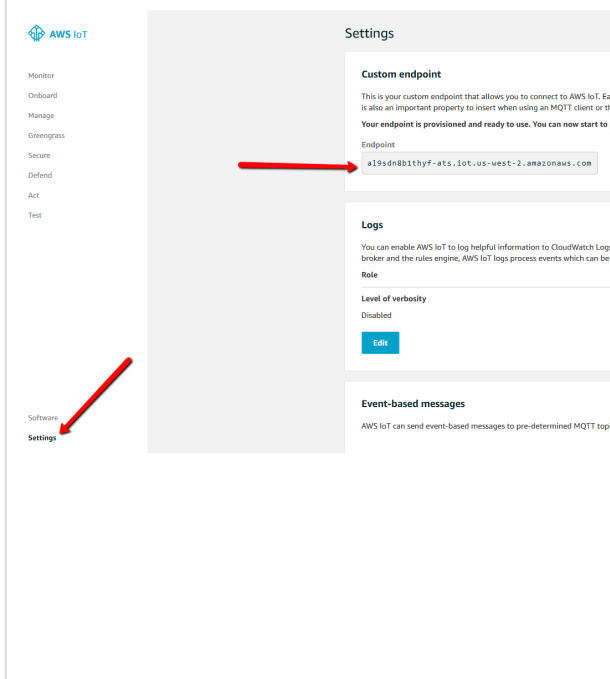
QoS Retain

Name

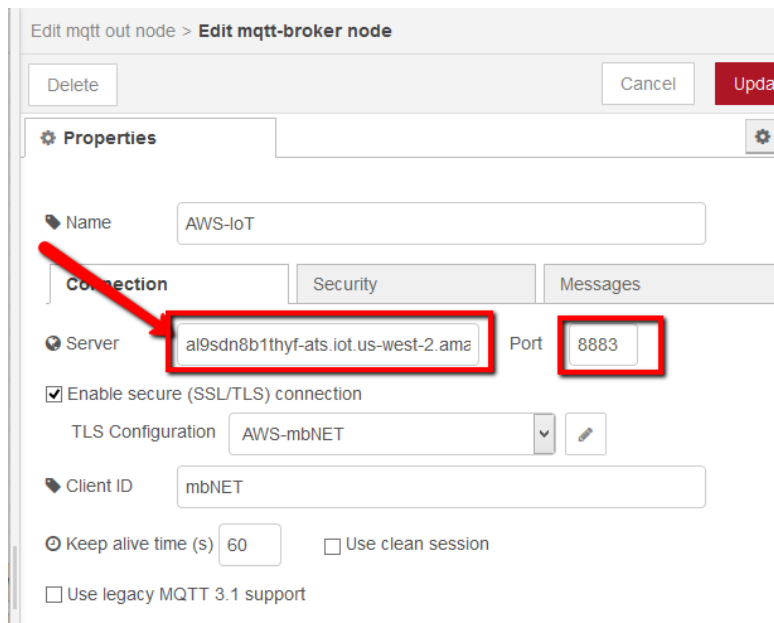
4.1 3.1 AWS endpoint settings

AWS	Node-RED
-----	----------

Select AWS console → Settings at AWS IoT



Edit your MQTT node settings and use the Endpoint from your AWS Endpoint setting for the MQTT server setting.



4.2 3.2. MQTT Client-ID settings

AWS	Node-RED
<p>The Things Name is your Client-ID for the MQTT node.</p>	

4.3 3.3. Import certificates

your Computer	Node-RED
<div style="border: 1px solid #ccc; padding: 5px;"> <p>your Computer</p> <ul style="list-style-type: none"> <input type="checkbox"/> Name d0377249f5-certificate.pem d0377249f5-private.pem.key d0377249f5-public.pem.key root_CA.pem </div>	<div style="border: 1px solid #ccc; padding: 5px;"> <p>Node-RED</p> <p>Edit mqtt out node > Edit mqtt-broker node > Edit tls-config node</p> <p>Delete Cancel Update</p> <p>Properties</p> <p><input type="checkbox"/> Use key and certificates from local files</p> <p>Certificate Upload d0377249f5-certificate.pem </p> <p>Private Key Upload d0377249f5-private.pem... </p> <p>Passphrase private key passphrase (optional)</p> <p>CA Certificate Upload root_CA.pem </p> <p><input checked="" type="checkbox"/> Verify server certificate</p> <p>Server Name for use with SNI</p> </div>

4.4 3.4. check connection

Change to AWS Test page and use the topic "mymbNETTopic" and subscribe.

Every Inject on "timestamp" will send a message to AWS and you can see it at the MQTT client test page.

AWS IoT MQTT client Connected as iotconsole-1588608133881-

Subscriptions

Subscribe to a topic

Publish to a topic

mymbNETTopic ✕

mymbNETTopic Export Clear Pause

Publish
Specify a topic and a message to publish with a QoS of 0.

Publish to topic

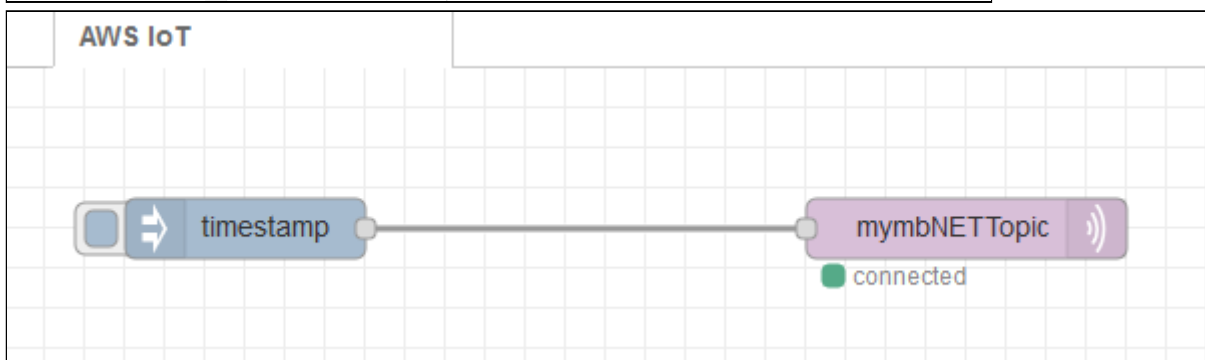
```

1 {
2   "message": "Hello from AWS IoT console"
3 }

```

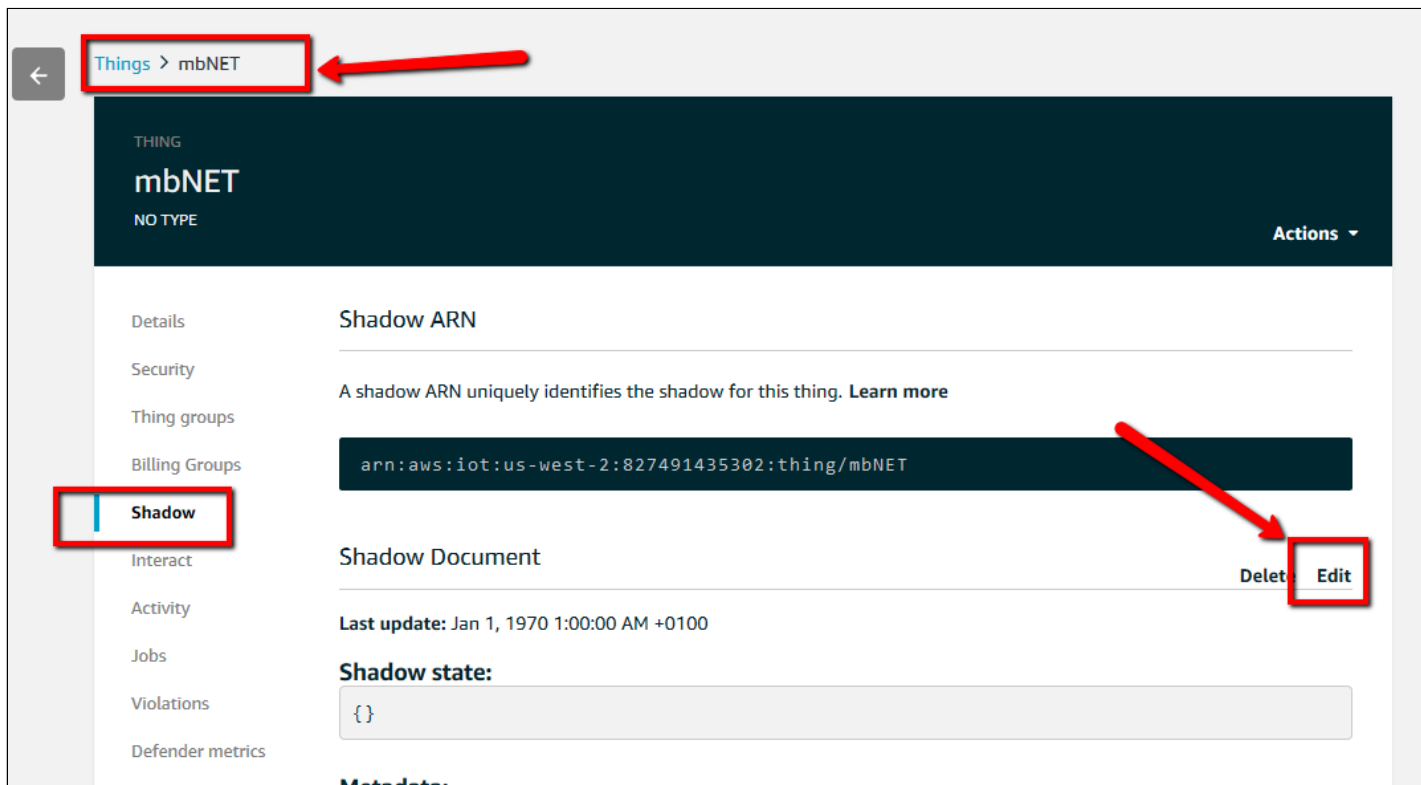
mymbNETTopic May 4, 2020 6:03:42 PM +0200 Export Hide

1588608219008

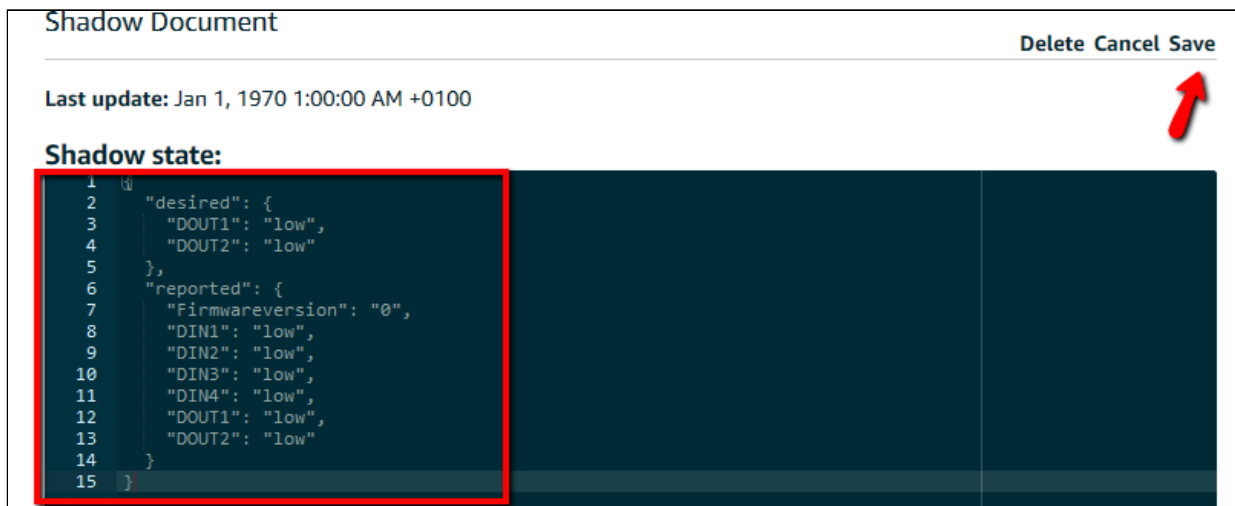


5 4. Advanced: Create and use AWS Shadow

5.1 4.1. Create our shadow document



Copy the Following code block into the shadow state:



```

Shadow Code

1  {
2  "desired": {
3    "DOUT1": "low",
4    "DOUT2": "low"
5  },
6  "reported": {
7    "Firmwareversion": "0",
8    "DIN1": "low",
9    "DIN2": "low",
10   "DIN3": "low",
11   "DIN4": "low",
12   "DOUT1": "low",
13   "DOUT2": "low"
14  }
15  }

```

5.2 4.2. Edit policy

Select your thing and then the certificate. Select then the policy and edit the policy

Things > mbNET > d0377249f5910b46ef95... > mbNET_allow_ALL

POLICY
mbNET_allow_ALL

Overview | Certificates | Versions | Groups | Non-compliance

Policy ARN
A policy ARN uniquely identifies this policy. [Learn more](#)
`arn:aws:iot:us-west-2:827491435302:policy/mbNET_allow_ALL`

Policy document
The policy document defines the privileges of the request. [Learn more](#)

Version 2 updated May 4, 2020 6:19:43 PM +0200 [Edit policy document](#)

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:us-west-2:827491435302:client/${iot:Connection.Thing.

```

Edit the policy as follows:

Policy

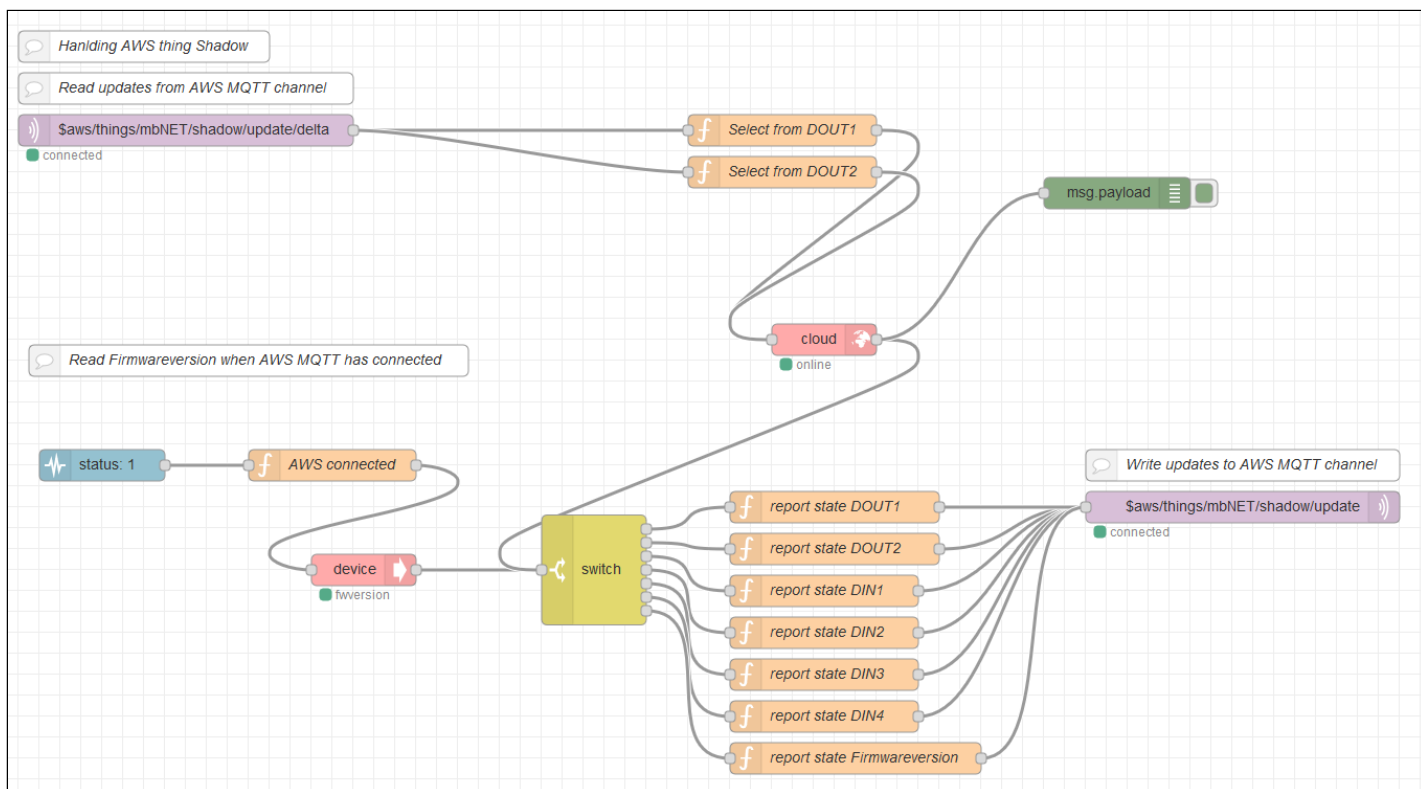
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:us-west-2:827491435302:client/${iot:Connection.Thing.ThingName}"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Publish",
      "Resource": [
        "arn:aws:iot:us-west-2:827491435302:topic/mymbNETTopic",
        "arn:aws:iot:us-west-2:827491435302:topic/${aws/things/${iot:Connection.Thing.ThingName}/shadow/update}"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": [
        "arn:aws:iot:us-west-2:827491435302:topic/mymbNETTopic",
        "arn:aws:iot:us-west-2:827491435302:topic/${aws/things/${iot:Connection.Thing.ThingName}/shadow/update/delta}"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": [
        "arn:aws:iot:us-west-2:827491435302:topicfilter/mymbNETTopic",
        "arn:aws:iot:us-west-2:827491435302:topicfilter/${aws/things/${iot:Connection.Thing.ThingName}/shadow/update/delta}",
        "arn:aws:iot:us-west-2:827491435302:topicfilter/${aws/things/${iot:Connection.Thing.ThingName}/shadow/update/get/accepted}",
        "arn:aws:iot:us-west-2:827491435302:topicfilter/${aws/things/${iot:Connection.Thing.ThingName}/shadow/update/get/rejected}",
        "arn:aws:iot:us-west-2:827491435302:topicfilter/${aws/things/${iot:Connection.Thing.ThingName}/shadow/update/documents}",
        "arn:aws:iot:us-west-2:827491435302:topicfilter/${aws/things/${iot:Connection.Thing.ThingName}/shadow/update/accepted}",
        "arn:aws:iot:us-west-2:827491435302:topicfilter/${aws/things/${iot:Connection.Thing.ThingName}/shadow/update/rejected}"
      ]
    }
  ]
}
```


⚠ Attention

The name "arn:aws:iot:us-west-2:827491435302" could be different to your setting. Use the name of the resource from "iot:connect" segment.

5.3 4.3 Add the Node-RED Flow

This Node-RED Flow is an example on how to interact between the mbNET router and AWS shadow. The "cloud" node communicates with the router to read and write the routers variables. In our case here we request to set/reset the digital Outputs 1 and/or 2 depending on the request from the AWS shadow. Whenever there is a change on this digital Outputs or Inputs we report to AWS shadow the status. The node "device" returns the status of the router interfaces and firmware version.



The flow can be downloaded here: <https://helpdesk.mbconnectline.com/en/index.php?type=page&urlcode=872841&title=NR10V1-AWS-Connector>

After activating the flow, you will see the firmwareversion of the mbNET in your AWS shadow:

mbNET
Actions ▾

NO TYPE

- Details
- Security
- Thing groups
- Billing Groups
- Shadow**
- Interact
- Activity
- Jobs
- Violations
- Defender metrics

Shadow ARN

A shadow ARN uniquely identifies the shadow for this thing. [Learn more](#)

`arn:aws:iot:us-west-2:827491435302:thing/mbNET`

Shadow Document

Last update: May 4, 2020 6:20:17 PM +0200

Shadow state:

```

{
  "desired": {
    "DOUT1": "low",
    "DOUT2": "low"
  },
  "reported": {
    "Firmwareversion": "6.2.2",
    "DIN1": "low",
    "DIN2": "low",
    "DIN3": "low",
    "DIN4": "low",
    "DOUT1": "low",
    "DOUT2": "low"
  }
}

```

[Delete](#) [Edit](#)

You can set the DOUT1 or DOUT2 (digital output) of the mbNET to "low" or "high" state. If digital Inputs are connected to the DI1-4 it will be reported here.