



MQTT

Connecting to the AWS, Amazon Web Services

Vers. 1.0 – Jul 2021

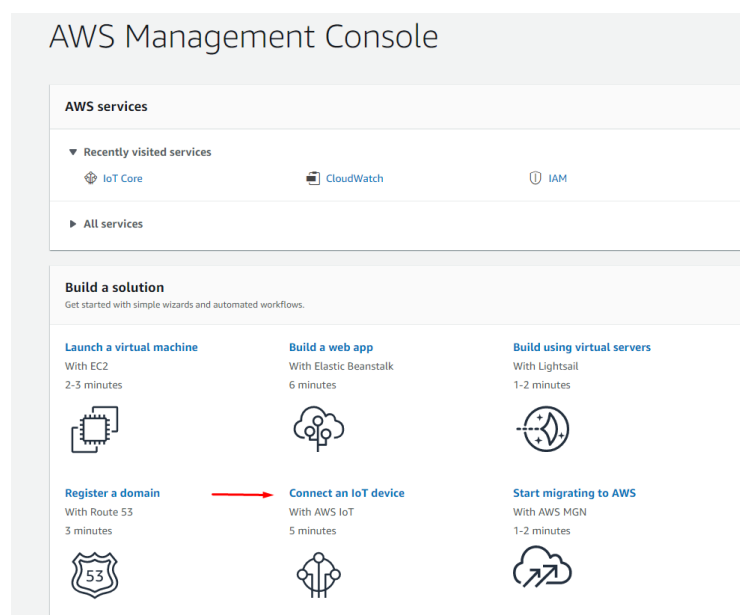
1. Introduction

All of Infinite's devices that support the MQTT protocol, are capable to connect to any local or remote MQTT Broker. Amazon Web Services is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis.

This document is a brief how-to guide for all device communications between Infinite's devices and the AWS IoT Hub which supports MQTT connectivity.

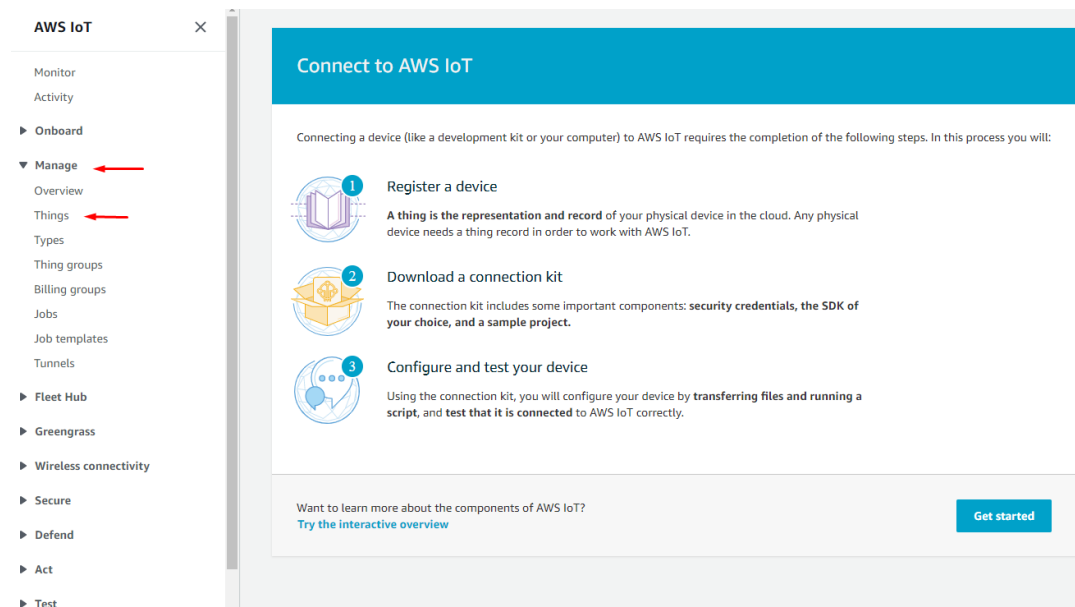
2. AWS Console

After creating an AWS account, navigate to the AWS Management Console page and click Connect an IoT device.

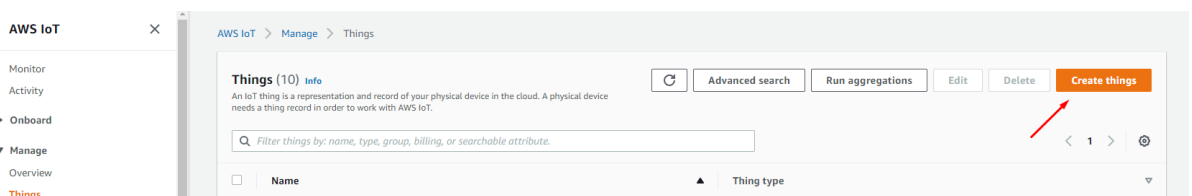


Open the Manage tab and click Things.

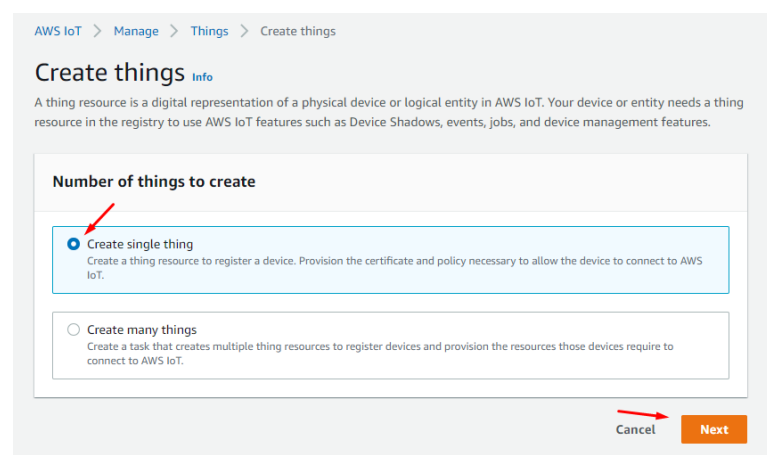
MQTT - Connecting to the AWS, Amazon Web Services



Click Create things.




Create a single thing.



MQTT - Connecting to the AWS, Amazon Web Services

Give the Thing a name.

Thing properties [info](#)

Thing name 

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
- ▶ Searchable thing attributes - optional
- ▶ Thing groups - optional
- ▶ Billing group - optional

Device Shadow [info](#)
Device Shadows allow connected devices to sync states with AWS. 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**

Auto-generate a new certificate. (AWS requires TLS communications)

Configure device certificate - optional [info](#)

A device requires a certificate to connect to AWS IoT. You can choose how you to register a certificate for your device now, or you can create and register a certificate for your device later. Your device won't be able to connect to AWS IoT until it has an active certificate with an appropriate policy.

Device certificate

 ☒ **Auto-generate a new certificate (recommended)**
Generate a certificate, public key, and private key using AWS IoT's certificate authority.

☐ Use my certificate
Use a certificate signed by your own certificate authority.

☐ Upload CSR
Register your CA and use your own certificates on one or many devices.

☐ Skip creating a certificate at this time
You can create a certificate for this thing and attach a policy to the certificate at a later time.

Cancel Previous **Next**

Create a policy to attach to the certificate.

Attach policies to certificate - optional [Info](#)

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

Policies (1) [Refresh](#) [Create policy](#)

Select up to 10 policies to attach to this certificate.

<input type="checkbox"/>	Name
<input type="checkbox"/>	nbiot

[Cancel](#) [Previous](#) [Create thing](#)

Name the policy and click advanced mode to define the types of actions that can be performed by our device.

Add statements Basic mode

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

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "",
6       "Action": "",
7       "Resource": ""
8     }
9   ]
10 }
```

[Add statement](#)

[Create](#)

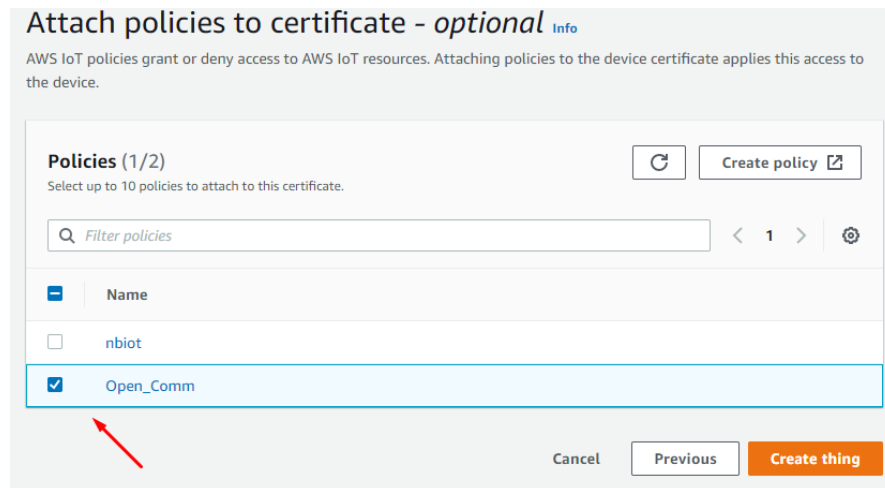
Delete the pre-existing statements and paste the following ones.

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

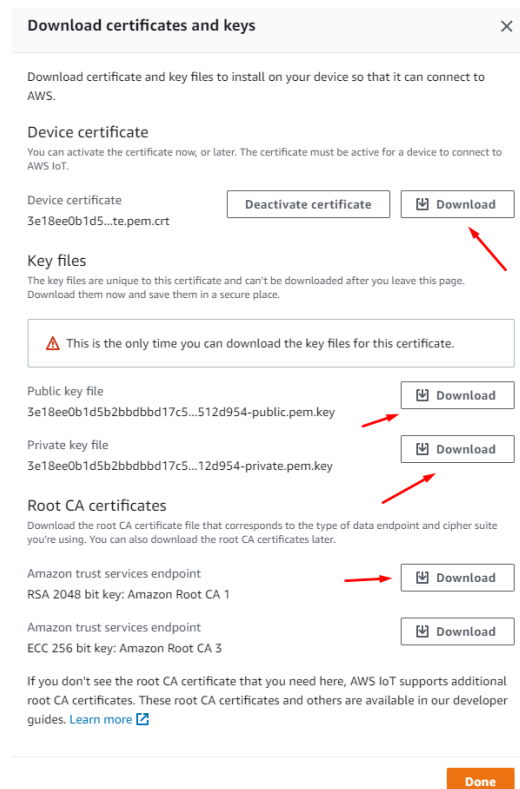
MQTT - Connecting to the AWS, Amazon Web Services

This policy is for testing purposes (it allows all communications to and from the device) and should be adjusted for your requirements.

Click refresh and choose the policy you just created and click Create thing.



In the windows that pops up you can download the certificates that were created.



3. Device Configuration with WA Manager

In the Edit Device window in WA Manager, tick the Use SSL box.

The screenshot shows the 'Edit Device' window for 'ADS-300'. The 'SSL Parameters' tab is active. The 'Use SSL' checkbox is checked, highlighted with a red arrow. Other configuration options include 'PSM Mode' (set to Off), 'RTC Correction' (0), 'UTC Time' (unchecked), and 'Offset' (0). The 'Device name' is 'ADS-300' and 'Unit ID' is '0'. A 'Comments' field at the bottom contains the text 'MOSQUITTO'.

Next, we configure the MQTT parameters.

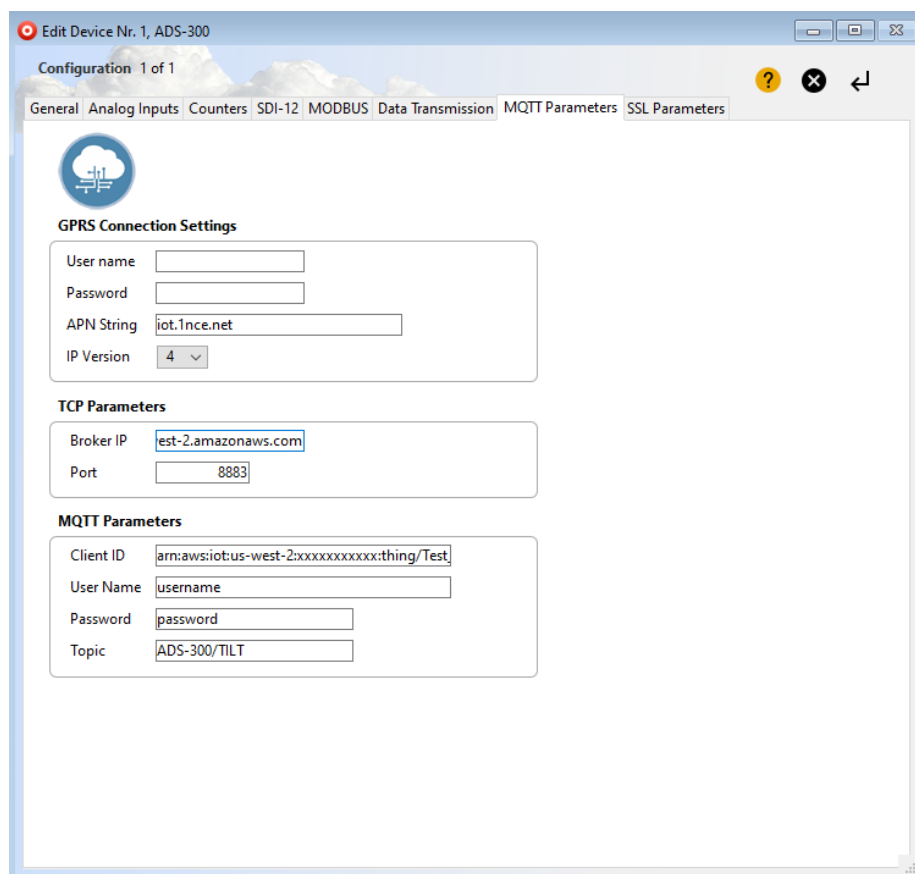
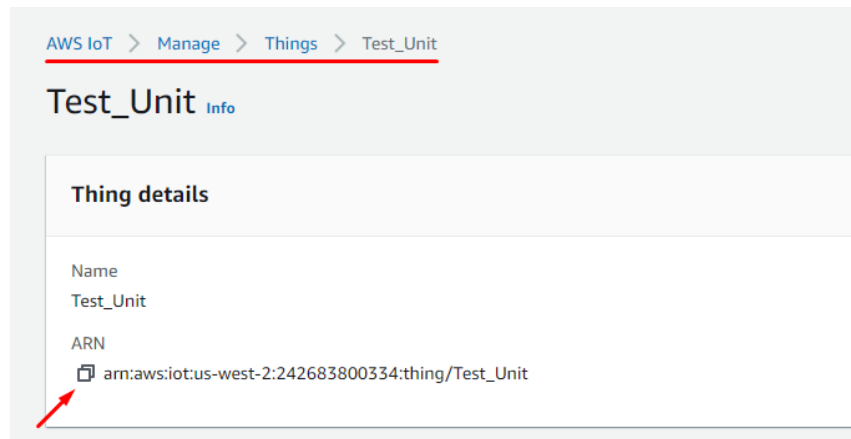
Although AWS supports MQTT connectivity, it is not a pure MQTT Broker and so it has some limitations regarding its MQTT parameters.

For the Broker IP, the Device data endpoint must be used that can be found in the AWS IoT Settings tab.

The screenshot shows the 'AWS IoT Settings' page. A notification at the top states 'Logging now supports JSON logs and fine-grained control.' Below this, the 'Settings' section is visible. The 'Device data endpoint' is highlighted with a red arrow. The endpoint URL is 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0u1v2w3x4y5z6.1.amazonaws.com'.

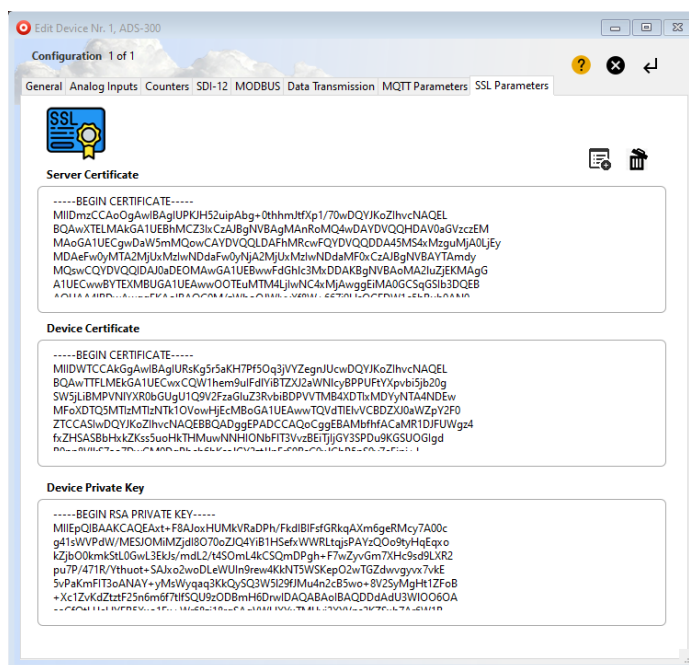
MQTT - Connecting to the AWS, Amazon Web Services

For the Client ID, the ARN (Amazon Resource Name) must be used that can be found in the Things tab.



Lastly, in the SSL Parameters tab, we copy and paste the three files needed for the TLS communication: Server Certificate (CA), Device Certificate and Device Private Key.

MQTT - Connecting to the AWS, Amazon Web Services



The Server Certificate is the Amazon trust services that you previously downloaded, the Device Certificate is the file you downloaded and the Device Private Key is the private key file. These files should be first opened with Notepad++ and their contents should be copy and pasted in the above tab. All files must be PEM formatted.

Your device can now securely connect to the AWS and send your encrypted telemetry data safely.

Disclaimer:

AWS, Amazon Web services is registered trademark of Amazon.com Inc, USA. All products and software mentioned in this document for educational and demonstration purposes.

Revision: 1.0

© 2021, Infinite Informatics Ltd

Infinite Informatics, Ltd

1, Valaoritou Street
GR-54626 Thessaloniki, Greece
Phone: +30-2310-553545
E: info@indinf.gr
W: www.infinite.com.gr