



MQTT

Connecting to Microsoft Azure

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. Microsoft Azure, commonly referred to as Azure, is a cloud computing service created by Microsoft for building, testing deploying, and managing applications and services through Microsoft-managed centers.

This document is a brief how-to guide for all device communications between Infinite's devices and Microsoft Azure.

2. Generating Self-Signed Device Certificate and Key

Azure requires TLS communications so we will have to create our own self-signed certificate and key for our device. We do that with the commercial-grade TLS toolkit openssl. The easiest way to do that is to simply install [git](#) on your computer and locate the openssl.exe file in this directory: `C:\Program Files\Git\usr\bin\openssl.exe`.

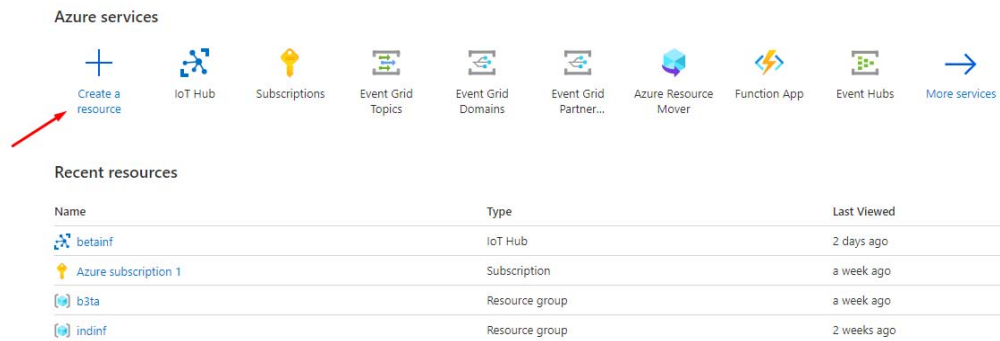
Open a Command Prompt or PowerShell window in the above directory and type the following commands to create the device certificate and key:

```
req -x509 -nodes -sha256 -days 365 -newkey rsa:2048 -keyout device.key -out device.crt - creates device certificate and private key
```

(These commands are for testing purposes and should be adjusted for different requirements.)

3. Creating an IoT Hub

After creating a Microsoft Account, the first step of this procedure is to create an IoT Hub. On the homepage of Microsoft Azure, click Create a resource and then search for IoT Hub.



Create an IoT Hub by filling out the project details and creating a new Resource group. You must choose East US as your region for the time being as Microsoft is working on enabling TLS1.2 on all regions.

[Home](#) > [Create a resource](#) > [IoT Hub](#) >

IoT hub

Microsoft

[Basics](#) [Networking](#) [Management](#) [Tags](#) [Review + create](#)

Create an IoT hub to help you connect, monitor, and manage billions of your IoT assets. [Learn more](#)

Project details
Choose the subscription you'll use to manage deployments and costs. Use resource groups like folders to help you organize and manage resources.

Subscription * ⓘ

Resource group * ⓘ [Create new](#)

Instance details

IoT hub name * ⓘ ✓

Region * ⓘ

[Review + create](#) [< Previous](#) [Next: Networking >](#)

Make sure that you configure the minimum TLS version as well.

IoT hub

Microsoft

Scale tier and units

Pricing and scale tier * ⓘ

F1: Free tier ▾

✘ Free IoT hubs are limited to one per subscription
[Learn how to choose the right IoT hub tier for your solution](#)

Number of F1 IoT hub units ⓘ

1

Determines how your IoT hub can scale. You can change this later if your needs increase.

Defender for IoT

Off

Turn on Defender for IoT and add an extra layer of threat protection to IoT Hub, IoT Edge, and your devices. [Learn more](#)

Pricing and scale tier ⓘ	F1	Device-to-cloud-messages ⓘ	Enabled
Messages per day ⓘ	8,000	Message routing ⓘ	Enabled
Cost per month	0.00 EUR	Cloud-to-device commands ⓘ	Enabled
Defender for IoT ⓘ	Disabled	IoT Edge ⓘ	Enabled
		Device management ⓘ	Enabled

Advanced settings

Scale

Device-to-cloud partitions ⓘ

2

Transport Layer Security (TLS)

Minimum TLS Version ⓘ

1.0

1.2

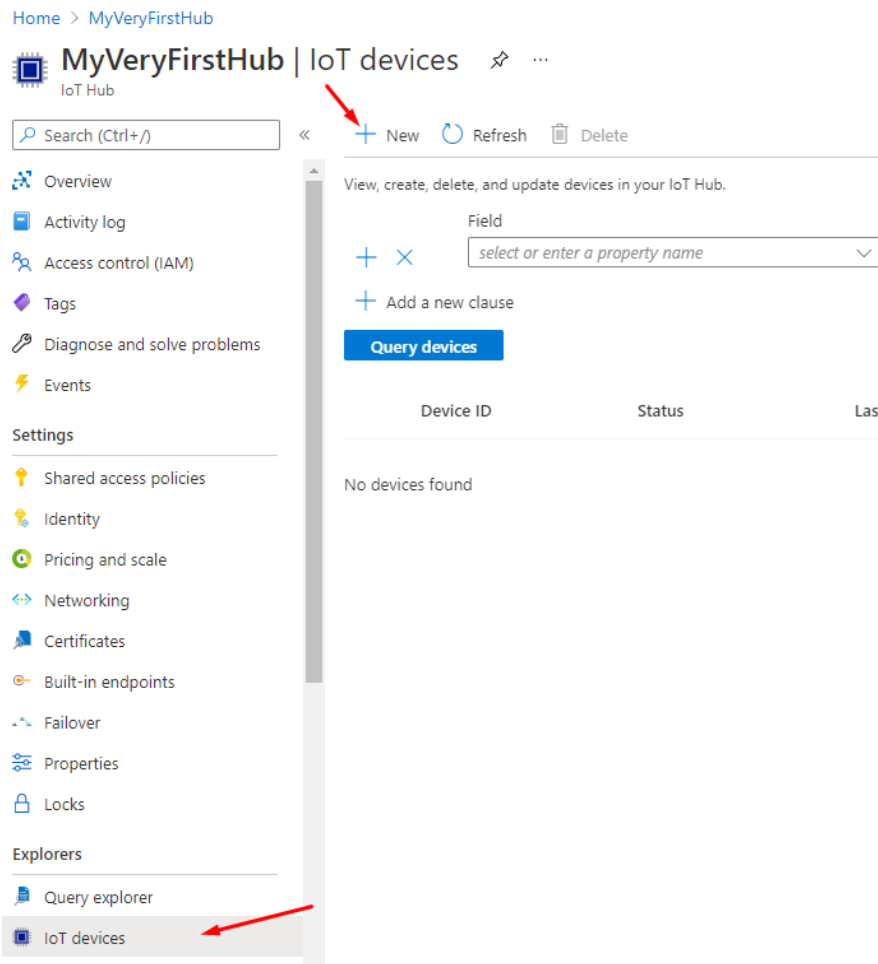
[Review + create](#)

[< Previous: Networking](#)

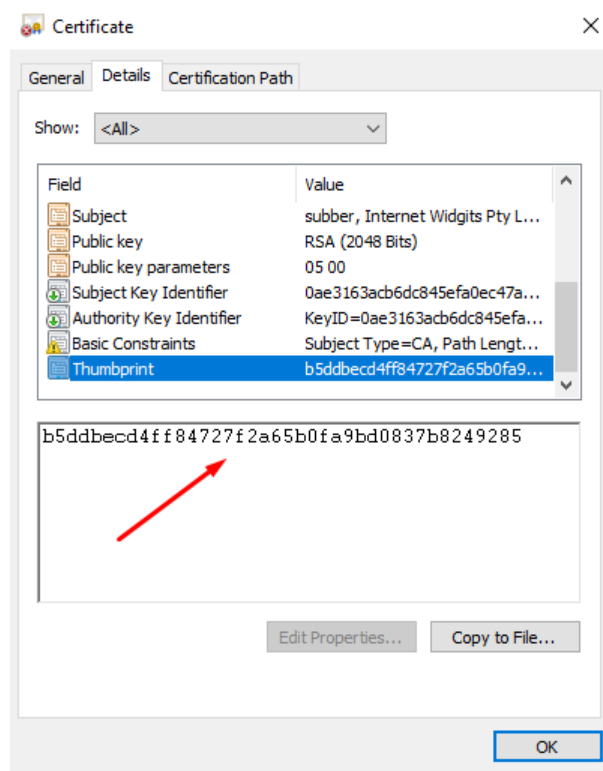
[Next: Tags >](#)

4. Creating a device

While you are on your IoT Hub page, enter the IoT devices tab and click New to create an IoT device.



On the next page name your device, choose X.509 Self-Signed as the authentication type and enter your certificates Primary Thumbprint. You can find this thumbprint by opening the device.crt file we created earlier.



[Home](#) > [MyVeryFirstHub](#) >

Create a device ...

Find Certified for Azure IoT devices in the Device Catalog

Device ID * ✓

Authentication type
 Symmetric key X.509 Self-Signed X.509 CA Signed

Primary Thumbprint * ✓

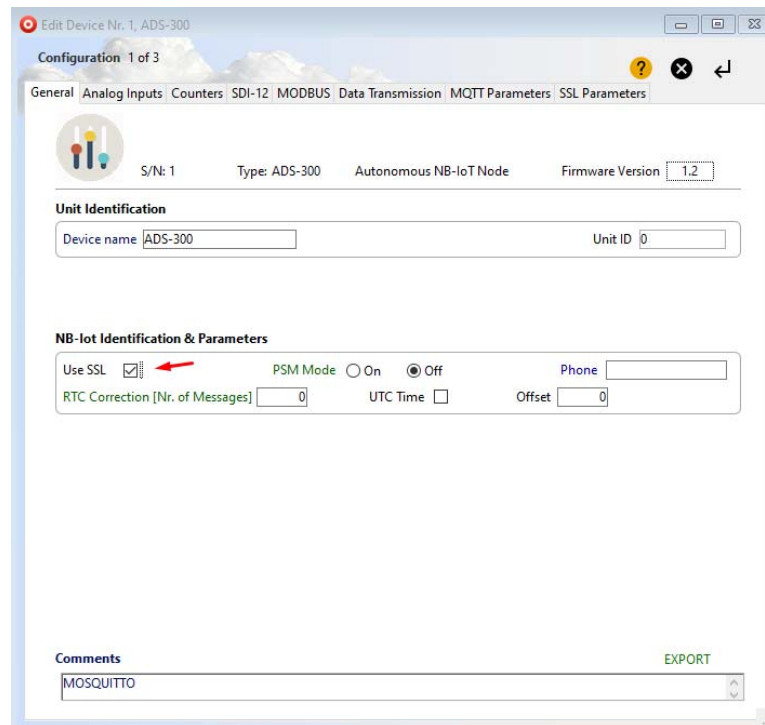
Secondary Thumbprint * ✓

Connect this device to an IoT hub
 Enable Disable

Parent device
No parent device
[Set a parent device](#)

5. Device Configuration with WA Manager

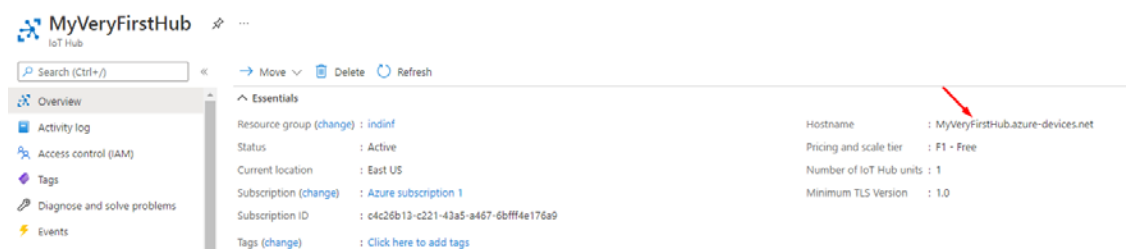
In the Edit Device window in WA Manager, tick the Use SSL box.



Next, we configure the MQTT parameters.

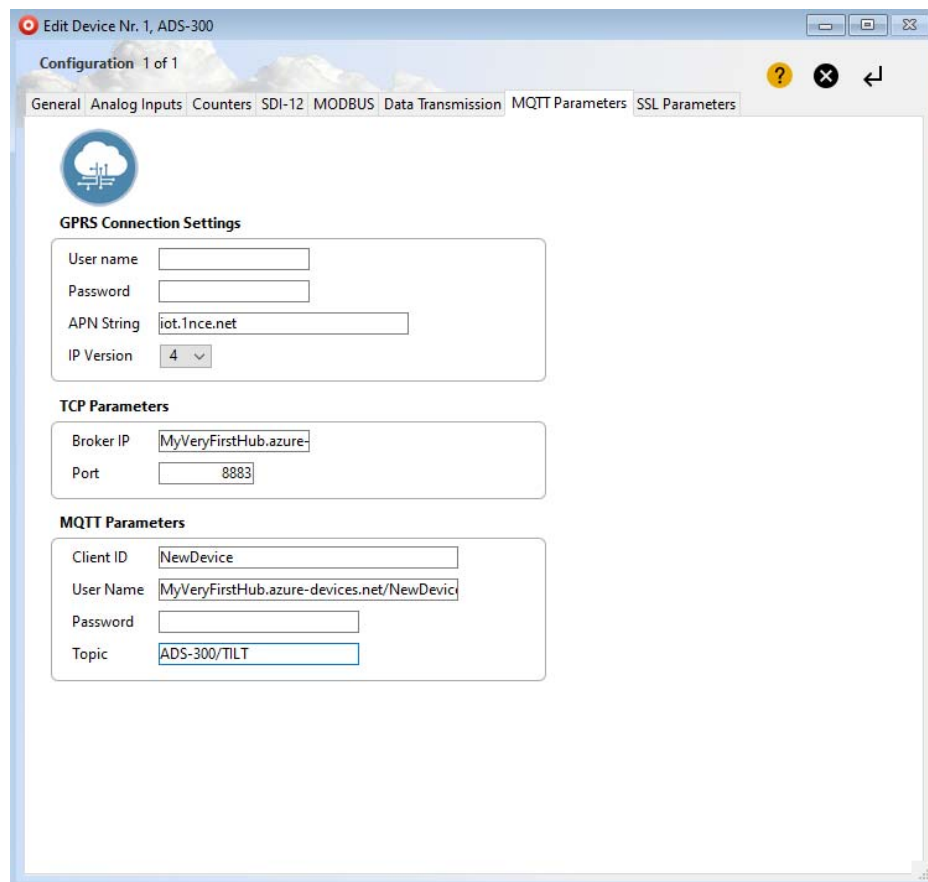
Although Azure supports MQTT connectivity, it is not a pure MQTT Broker and so it has some limitations regarding its MQTT parameters.

For the Broker IP, the IoT Hub endpoint must be used that can be found in the IoT Hub page.



For the Client ID, the Device ID must be used that we used to create our device.

MQTT - Connecting to Microsoft Azure

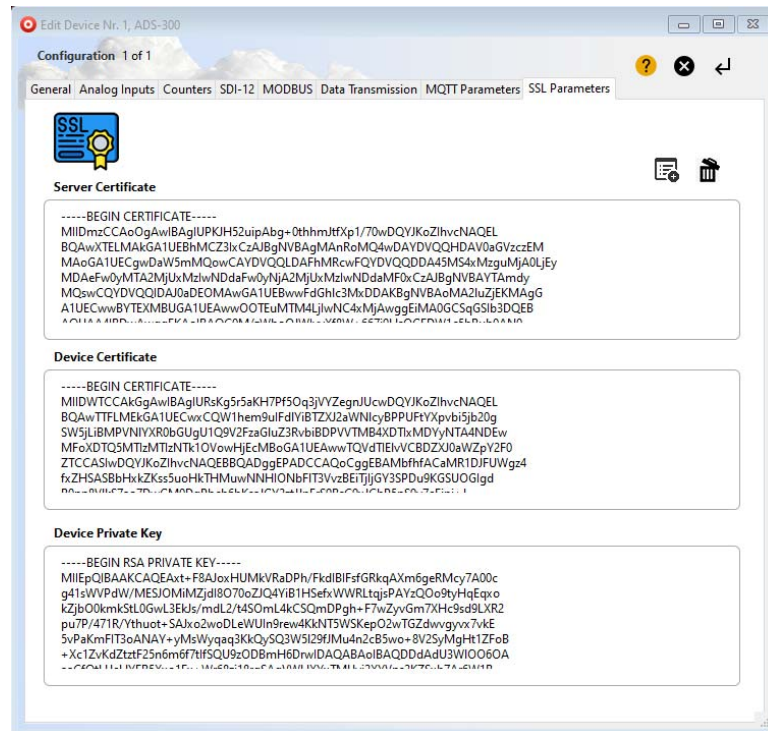


The screenshot shows a software interface for configuring a device. The window title is 'Edit Device Nr. 1, ADS-300'. The 'MQTT Parameters' tab is selected. The configuration is divided into three sections:

- GPRS Connection Settings:**
 - User name:
 - Password:
 - APN String:
 - IP Version:
- TCP Parameters:**
 - Broker IP:
 - Port:
- MQTT Parameters:**
 - Client ID:
 - User Name:
 - Password:
 - Topic:

The username must be of this format based on the name of our DeviceID and IoT Hub name: MyVeryFirstHub.azure-devices.net/NewDevice/?api-version=2018-06-30

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.



The Server Certificate is [this](#) Digicert CA, the Device Certificate is the device.crt file and the Device Private Key is the device.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 connect Azure and send your encrypted data safely.

Disclaimer:

Azure is a registered trademark of Microsoft Corporation, 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