Miromico miro Edge gateway: Getting Started Guide for AWS IoT Core for LoRaWAN

2020.12.22.0

Table of Contents

1 Document Information

1.1 Naming Conventions

The term “downlink device” or “endpoint device” is used in this document to refer to a LoRaWAN device that connects to a LoRaWAN “Gateway”. The “Gateway” in turn, connects to AWS IoT Core for LoRaWAN.

1.2 Revision History (Version, Date, Description of change)

V1.0 Release 2021-September-07

2 Overview

This document describes how to set up a miro Edge gateway to connect to AWS IoT Core for LoRaWAN.

3 Hardware Description

3.1 Datasheet

Technical datasheet:
https://docs.miromico.ch/datasheets/_attachments/gateway/miro_Edge_datasheet_V1_0.pdf

General documentation:
https://docs.miromico.ch/miroEdge/

4 Setup your AWS account and Permissions

If you don’t have an AWS account, refer to the instructions in the guide here. The relevant sections are **Sign up for an AWS account** and **Create a user and grant permissions**.

4.1 Overview

The high-level steps to get started with AWS IoT Core for LoRaWAN are as follows:

1. Set up Roles and Policies in IAM
2. Add a Gateway (see section Add the Gateway to AWS IoT)
3. Add Device(s) (see section Add a LoRaWAN Device to AWS IoT)
    a. Verify device and service profiles
    b. Set up a Destination to which device traffic will be routed and processed by a rule.

These steps are detailed below. For additional details, refer to the AWS LoRaWAN developer guide.

4.2 Set up Roles and Policies in IAM

4.2.1 Add an IAM Role for CUPS server

Add an IAM role that will allow the Configuration and Update Server (CUPS) to handle the wireless gateway credentials.

This procedure needs to be done only once, but must be performed before a LoRaWAN gateway tries to connect with AWS IoT Core for LoRaWAN.

- Go to the IAM Roles page on the IAM console
- Choose **Create role**.
- On the **Create Role** page, choose **Another AWS account**.
- For **Account ID**, enter your account id.
- Choose **Next: Permissions**
- In the search box next to **Filter policies**, enter *AWSIoTWirelessGatewayCertManager*.

- If the search results show the policy named *AWSIoTWirelessGatewayCertManager*, select it by clicking on the checkbox.
- If the policy does not exist, please create it as follows:
  - Go to the IAM console
  - Choose **Policies** from the navigation pane.
  - Choose **Create Policy**. Then choose the **JSON** tab to open the policy editor. Replace the existing template with this trust policy document:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "IoTWirelessGatewayCertManager",
            "Effect": "Allow",
            "Action": [
                "iot:CreateKeysAndCertificate",
                "iot:DescribeCertificate",
                "iot:ListCertificates",
                "iot:RegisterCertificate"
            ],
            "Resource": "*"
        }
    ]
}
```

  - Choose **Review Policy** to open the *Review* page.
  - For **Name**, enter *AWSIoTWirelessGatewayCertManager.* **Note** that you must enter the name as AWSIoTWirelessGatewayCertManager and must not use a different name. This is for consistency with future releases.
  - For **Description**, enter a description of your choice.
  - Choose **Create policy**. You will see a confirmation message showing the policy has been created.

- Choose **Next: Tags**, and then choose **Next: Review**.
- In **Role name**, enter *IoTWirelessGatewayCertManagerRole*, and then choose **Create role**.
  - **Note** that you must not use a different name. This is for consistency with future releases.
- In the confirmation message, choose **IoTWirelessGatewayCertManagerRole** to edit the new role.
- In the **Summary**, choose the **Trust relationships** tab, and then choose **Edit trust relationship**.
- In the **Policy Document**, change the **Principal** property to represent the IoT Wireless service:

```
"Principal": {
    "Service": "iotwireless.amazonaws.com"
},
```

After you change the Principal property, the complete policy document should look like this:

```
  {
    "Version": "2012-10-17",
    "Statement": [
      {
          "Effect": "Allow",
          "Principal": {
            "Service": "iotwireless.amazonaws.com"
          },
          "Action": "sts:AssumeRole",
          "Condition": {}
      }
    ]
    }
```

- Choose **Update Trust Policy** to save your changes and exit.

At this point, you've created the *IoTWirelessGatewayCertManagerRole* and you won't need to do this again.

NOTE *– The examples in this document are intended only for dev environments. All devices in your fleet must have credentials with privileges that authorize only intended actions on specific resources. The specific permission policies can vary for your use case. Identify the permission policies that best meet your business and security requirements. For more information, refer to Example policies and Security Best practices.*

4.2.2 Add IAM role for Destination to AWS IoT Core for LoRaWAN

Prepare your AWS account to work with AWS IoT Core for LoRaWAN.

Create a policy that gives the role permissions to describe the IoT endpoint and publish messages to AWS IoT.

- Go to the IAM console
- Choose **Policies** from the navigation pane.
- Choose **Create Policy**. Then choose the **JSON** tab to open the policy editor. Replace the existing template with this trust policy document:

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

- Choose **Review Policy** to open the Review page. For Name, enter a name of your choice. For **Description**, enter a description of your choice.
- Choose **Create policy**. You will see a confirmation message indicating that the policy has been created.

Now create the Role:

- In the IAM console, choose **Roles** from the navigation pane to open the **Roles** page.
- Choose **Create Role**.
- In **Select type of trusted entity**, choose **Another AWS account**.
- In **Account ID**, enter your AWS account ID, and then choose **Next: Permissions**.
- Search for the IAM policy you just created by entering the policy name in the search bar.
- In the search results, select the checkbox corresponding to the policy
- Choose **Next: Tags**.
- Choose **Next: Review** to open the Review page.
- For **Role name**, enter an appropriate name of your choice. For **Description**, enter a description of your choice.
- Choose **Create role**. You will see a confirmation message indicating that your role has been created.

Update your role's trust relationship to grant AWS IoT Core for LoRaWAN permission to assume this IAM role when delivering messages from devices to your account

- In the IAM console, choose **Roles** from the navigation pane to open the **Roles** page

- Enter the name of the role you created earlier in the search window, and click on the role name in the search results. This opens up the Summary page.
- Choose the **Trust relationships** tab to navigate to the Trust relationships page.
- Choose **Edit trust relationship**. The principal AWS role in your trust policy document defaults to root, and must be changed. Replace the existing policy with this:

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Allow",
        "Principal": {
            "Service": "iotwireless.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {}
      }
    ]
}
```

- Choose **Update Trust Policy.** Under **Trusted entities**, you will see: *The identity provider(s) iotwireless.amazonaws.com*.

4.3 Add the Gateway to AWS IoT

4.3.1 Preparation

To complete setting up your gateway, you need:

- LoRaWAN region. For example, if the gateway is deployed in a US region, the gateway must support LoRaWAN region US915.
- Gateway LNS-protocols. Currently, the LoRa Basics Station protocol is supported.
- Gateway ID (**GatewayEUI**). This is used to establish the connection between the LNS and the gateway.
    - To determine the GatewayEUI, look at the MAC address (label on back of gateway housing). The EUI is constructed by adding **FFFE** in the middle of the mac address.
    - Example:
      MAC:010203040506
      GatewayEUI: 010203**FFFE**040506

4.3.2 Add the LoRaWAN Gateway

To register the Gateway with AWS IoT Core for LoRaWAN, follow these steps:

- Go to the AWS IoT console.
- Select **Wireless connectivity** in the navigation panel on the left.
- Choose **Intro**, and then choose **Get started**. This step is needed to pre-populate the default profiles.
- Under **Add LoRaWAN gateways and wireless devices**, choose **Add gateway**.
- In the **Add gateway** section, fill in the **GatewayEUI** (see Section 4.3.1) and **Frequency band (RF Region)** fields.
- Enter a descriptive name in the **Name – optional** field. We recommend that you use the GatewayEUI as the name.
- Choose **Add gateway**
- On the **Configure your Gateway** page, find the section titled **Gateway certificate**.
- Select **Create certificate**.
- Once the **Certificate created and associated with your gateway** message is shown, select **Download certificates** to download the certificate (xxxxx.cert.pem) and private key (xxxxxx.private.key).
- In the section **Provisioning credentials**, choose **Download server trust certificates** to download the CUPS (cups.trust) and LNS (lns.trust) server trust certificates.

- Copy the CUPS and LNS endpoints and save them for use while configuring the gateway.
- Choose **Submit** to add the gateway.

4.4 Add a LoRaWAN Device to AWS IoT

4.4.1 Preparation

Locate and note the following specifications about your endpoint device.

- LoRaWAN region. This must match the gateway LoRaWAN region. The following Frequency bands (RF regions) are supported:
  - EU868
  - US915
  - EU433
- MAC Version. This must be one of the following:
  - V1.0.2
  - v1.0.3
  - v1.1
- OTAA v1.0x and OTAA v1.1 are supported.
- ABP v1.0x and ABP v1.1 are supported.

Locate and note the following information from your device manufacturer:

- For OTAA v1.0x devices: DevEUI, AppKey, AppEUI
- For OTAA v1.1 devices: DevEUI, AppKey, NwkKey, JoinEUI
- For ABP v1.0x devices: DevEUI, DevAddr, NwkSkey, AppSkey
- For ABP v1.1 devices: DevEUI, DevAddr, NwkSEnckey, FNwkSIntKey, SNwkSIntKey, AppSKey

4.4.2 Verify Profiles

AWS IoT Core for LoRaWAN supports device profiles and service profiles. Device profiles contain the communication and protocol parameter values the device needs to communicate with the network server. Service profiles describe the communication parameters the device needs to communicate with the application server.

Some pre-defined profiles are available for device and service profiles. Before proceeding, verify that these profile settings match the devices you will be setting up to work with AWS IoT Core for LoRaWAN.

- Navigate to the AWS IoT console. In the navigation pane, choose **Wireless connectivity.**
- In the navigation pane, choose **Profiles**
- In the **Device Profiles** section, there are some pre-defined profiles listed.
- Check each of the profiles to determine if one of them will work for you.
- If not, select **Add device profile** and set up the parameters as needed. For US 915 as an example, the values are:
  - MacVersion 1.0.3
  - RegParamsRevision RP002-1.0.1
  - MaxEirp 10
  - MaxDutyCycle 10
  - RfRegion US915
  - SupportsJoin true
- Continue once you have a device profile that will work for you.
- In the **Service Profiles** section, there are some pre-defined profiles listed. Check each of the profiles to determine if one of them will work for you.
- If not, select **Add service profile** and set up the parameters as needed. As an example, the default service profile parameters are shown below. However, only the AddGwMetadata setting can be changed at this time.
  - UlRate 60

UlBucketSize	4096
DlRate	60
DlBucketSize	4096
AddGwMetadata	true
DevStatusReqFreq	24
DrMax	15
TargetPer	5
MinGwDiversity	1

Proceed only if you have a device and service profile that will work for you.

4.4.3 Set up a Destination for device traffic

Because most LoRaWAN devices don't send data to AWS IoT Core for LoRaWAN in a format that can be consumed by AWS services, traffic must first be sent to a Destination. A Destination represents the AWS IoT rule that processes a device's data for use by AWS services. This AWS IoT rule contains the SQL statement that selects the device's data and the topic rule actions that send the result of the SQL statement to the services that will use it.

For more information on Destinations, refer to the AWS LoRaWAN developer guide.

A destination consists of a Rule and a Role. To set up the destination:

- Navigate to the AWS IoT console. In the navigation pane, choose **Wireless connectivity,** and then **Destinations**
- Choose **Add Destination**
- On the **Add destination** page, in the **Permissions** section select the IAM role you had created earlier, from the drop-down.
- Under **Destination details** enter *ProcessLoRa* as the **Destination name**, and an appropriate description under **Destination description – optional.**

  NOTE: The Destination name can be anything. For getting started and consistency, choose *ProcessLoRa* for the first integration with AWS IoT Core for LoRaWAN.

- For **Rule name** enter *LoRaWANRouting*. Ignore the section **Rules configuration – Optional** for now. The Rule will be set up later in the "Hello World" sample application – see Create the IoT Rule for the destination
- Choose **Add Destination.** You will see a message "*Destination added*", indicating the destination has been successfully added.

4.4.4 Register the Device

Now register an endpoint device with AWS IoT Core for LoRaWAN as follows:

- Go to the AWS IoT console.
- Select **Wireless connectivity** in the navigation panel on the left.
- Select **Devices**
- Choose **Add wireless device**
- On the **Add device** page, select the LoRaWAN specification version in the drop-down under **Wireless device specification**.
- Under **LoRaWAN specification and wireless device configuration**, enter the **DevEUI** and confirm it in the **Confirm DevEUI** field.
- Enter the remaining fields as per the OTAA/ABP choice you made above.
- Enter a name for your device in the **Wireless device name – optional** field.
- In the **Profiles** section, under **Wireless device profile**, find a drop-down option that corresponds to your device and region.
  - NOTE: Compare your device details to ensure the device profile is correct. If there are no valid default options, you will have to create a new profile (see the section Verify Profiles).
- Choose **Next**

- Choose the destination you created earlier (*ProcessLoRa*) from the drop-down under **Choose destination.**
- Choose **Add device**
- You will see a message saying "*Wireless device added*", indicating that your device has been set up successfully.

5 Set up the Gateway

5.1 Set up Gateway hardware

- Screw in the **LoRa® antenna** and the flat **LTE antenna (WWAN)**
- Connect an **ethernet** cable, either directly to your PC or to a switch/router in your network (see steps in Section 5.2)
- Connect the provided USB power supply (micro USB **DC-IN**). Alternatively, it can be powered by a switch supporting power-over-ethernet.
- Optionally, cellular LTE connection can be used, see https://docs.miromico.ch/miroEdge/bics_registration.html and https://docs.miromico.ch/miroEdge/no_sim.html

5.2 Set up Gateway Software

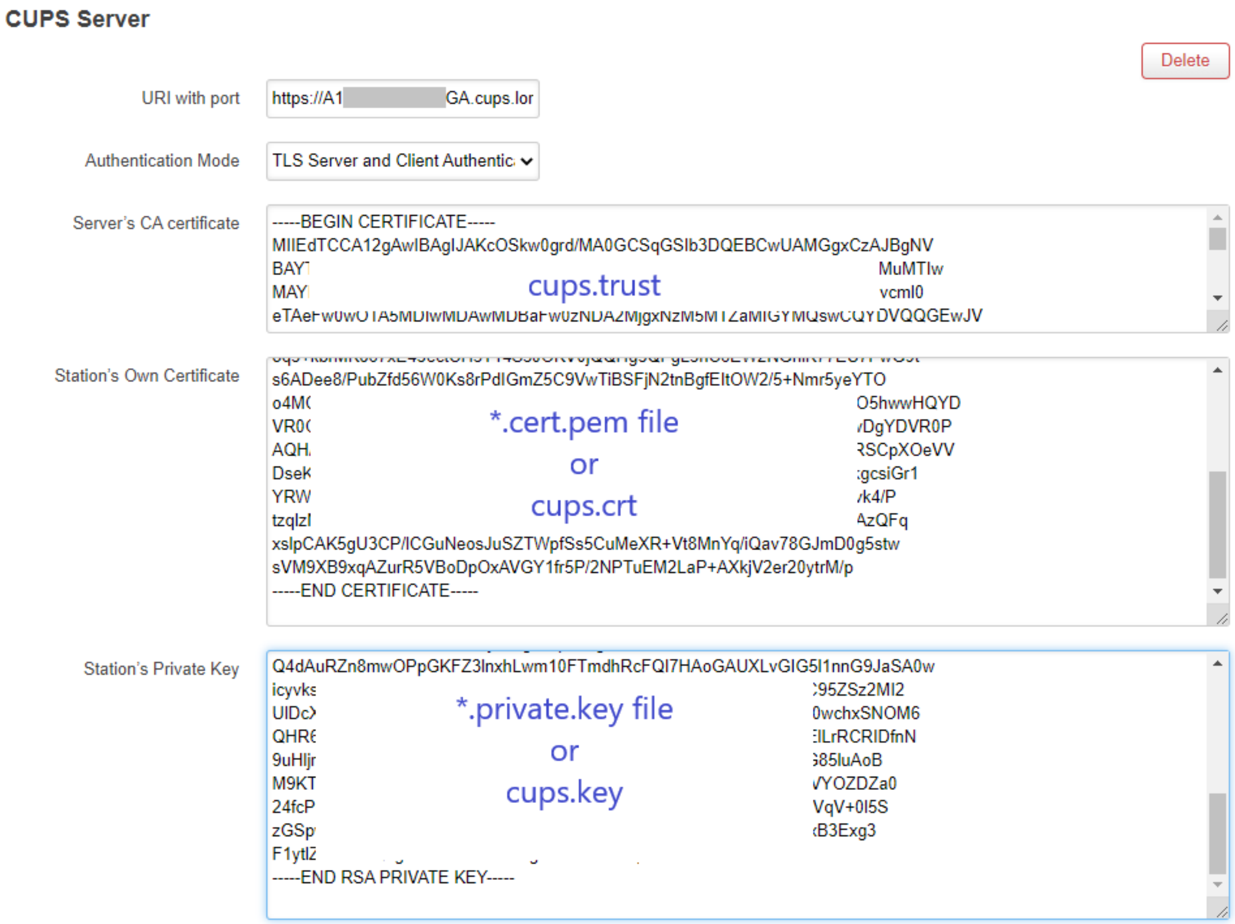
The miro Edge gateway can be configured over its web frontend. It can be accessed directly by entering its IP address in a web browser (address assigned by a DHCP server in your network). Alternatively, the gateway can be connected directly via ethernet to your PC: For that, follow the steps described under https://docs.miromico.ch/miroEdge/configuration.html, until you are **logged in** and then follow the steps in the next Section 5.3.

5.3 Configure the Gateway device

The following steps in the web frontend are required to configure the gateway:

- Select **LoRaWAN** under the menu option **Services**
  - Select **LoRa Basics™ Station** in the dropdown menu for **forwarder type**. If the option is not available, update the gateway to the newest firmware version: https://docs.miromico.ch/miroEdge/update.html
  - Click **Save and Apply**
- Select **LoRaWAN Basicstation** under the menu option **Services**
- Under **CUPS Server**, click **Add**
- Fill in the CUPS server settings (see Figure 1 CUPS Server settings):
  - Enter the **URI with port** (CUPS endpoint from Section 4.3.2) e.g. **https://XXXXXXX.cups.lorawan.us-east-1.amazonaws.com:443**
  - Under **Authentication Mode** select **TLS Server and Client Authentication**
  - Open the **cups.trust** file in a text editor (see Section 4.3.2) and copy-paste its contents into the **Server's CA certificate** field
  - Open the ***.cert.pem** file in a text editor (see Section 4.3.2) and copy-paste its contents into the **Station's Own Certificate** field
  - Open the ***.private.key** file in a text editor (see Section 4.3.2) and copy-paste its contents into the **Station's Private Key** field
- Under **LNS Server**, click **Add**
- Fill in the LNS server settings (see Figure 2 LNS Server settings):
  - Enter the **URI with port**, (LNS endpoint from Section 4.3.2) e.g. **wss:// XXXXXXX.lns.lorawan.us-east-1.amazonaws.com:443**
  - Under **Authentication Mode** select **TLS Server Authentication**
  - Open the **lns.trust** file in a text editor (see Section 4.3.2) and copy-paste its contents into the **Server's CA certificate** field
- Click **Save and Apply**

- It is recommended to change the router's default password under **System->Administration**



CUPS Server

URI with port: https://A1[redacted]GA.cups.lor

Authentication Mode: TLS Server and Client Authentic

Server's CA certificate:

```
-----BEGIN CERTIFICATE-----
MIIEdTCCA12gAwIBAgIJAKcOSkw0grd/MA0GCSqGSIb3DQEBCwUAMGgxCzAJBgNV
BAY[redacted]MuMTIw
MAY[redacted]vcml0
eTAeFw0wOTA5MDIwMDAwMDBaFw0zNDA2MjgxNzM5MTZaMIGYMQswCQYDVQQGEwJV
```

cups.trust

Station's Own Certificate:

```
s6ADee8/PubZfd56W0Ks8rPdIGmZ5C9VwTiBSFjN2tnBgfEltOW2/5+Nmr5yeYTO
o4M[redacted]O5hwwHQYD
VR0[redacted]DgYDVR0P
AQH[redacted]RSCpXOeVV
Dse[redacted]gcsiGr1
YRW[redacted]k4/P
tzqlz[redacted]AzQFq
xslpCAK5gU3CP/ICGuNeosJuSZTWpfSs5CuMeXR+Vt8MnYq/iQav78GJmD0g5stw
sVM9XB9xqAZurR5VBoDpOxAVGY1fr5P/2NPTuEM2LaP+AXkjV2er20ytrM/p
-----END CERTIFICATE-----
```

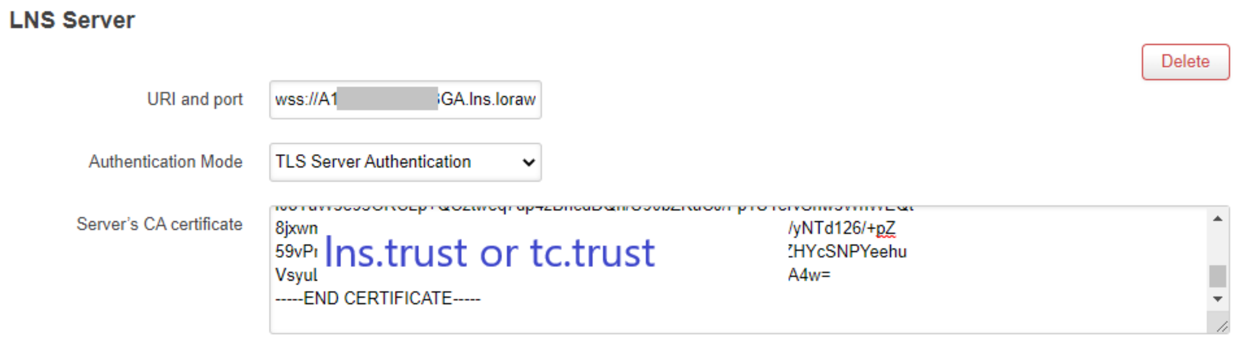
*.cert.pem file
or
cups.crt

Station's Private Key:

```
Q4dAuRZn8mwOPpGKFZ3lnxhLwm10FTmdhRcFQI7HAoGAUXLvGIG5I1nnG9JaSA0w
icyvks[redacted]95ZSz2MI2
UIDc[redacted]0wchxSNOM6
QHR[redacted]ILrRCRIDfnN
9uHljr[redacted]85luAoB
M9KT[redacted]YOZDZa0
24fcP[redacted]VqV+0I5S
zGSp[redacted]B3Exg3
F1ytlZ[redacted]
-----END RSA PRIVATE KEY-----
```

*.private.key file
or
cups.key

Figure 1 CUPS Server settings



LNS Server

URI and port: wss://A1[redacted]GA.lns.loraw

Authentication Mode: TLS Server Authentication

Server's CA certificate:

```
8jxwn[redacted]/yNTd126/+pZ
59vP[redacted]HYcSNPYeehu
Vsyul[redacted]A4w=
-----END CERTIFICATE-----
```

lns.trust or tc.trust

Figure 2 LNS Server settings

6 Add End Device(s)

The process of adding end devices is described in Section 4.4

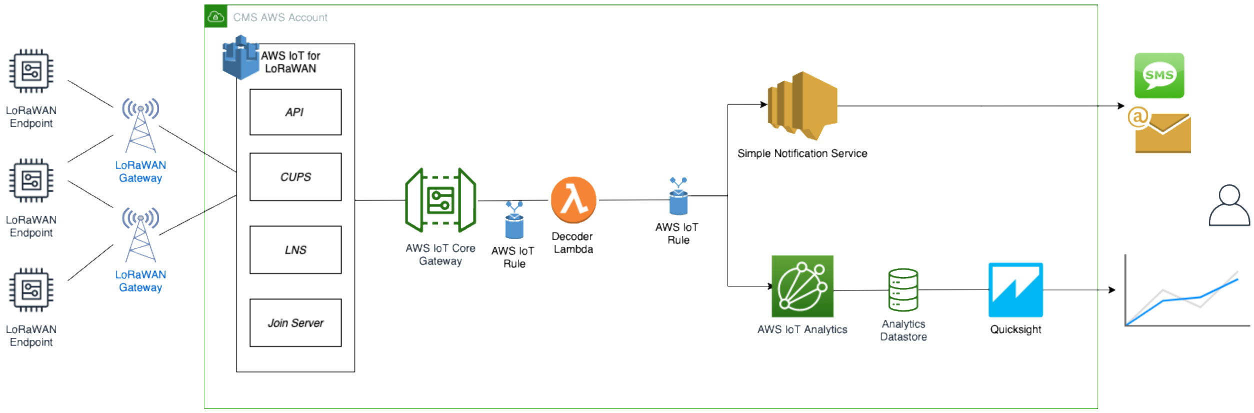
7 Verifying Operation – a "Hello World" example

To verify that the gateway connected to AWS:

- Open https://console.aws.amazon.com/iot/home
- Click on **Wireless connectivity**, then on **Gateways**
- Under **Last uplink received** a recent date and time should be displayed for the Gateway ID you configured

Once setup is completed, provisioned OTAA devices can join the network and start to send messages. Messages from devices can then be received by AWS IoT Core for LoRaWAN and forwarded to the IoT Rules Engine.

Instructions for a sample Hello World application are given below, assuming that the device has joined and is capable of sending uplink traffic. The architecture for this sample application is:



7.1 Create lambda function for destination rule

Create the lambda function to process device messages processed by the destination rule.

- Go to the AWS Lambda console (console.aws.amazon.com/lambda).
- Click on **Functions** in the navigation pane
- Click on **Create function**
- Select **Author from scratch.** Under Basic information, enter the function name "*sailboatdecoder*" and choose *Runtime Node.js 12.x*. from the drop-down under **Runtime**.
- Click on **Create function**.
- Navigate to *<<provide your github repository URL>>* and copy the code for the lambda function.
- Under **Function code**, paste the copied code into the editor under the **index.js** tab.
- Once the code has been pasted, choose "**Deploy**" to deploy the lambda code.
- Click on the **Permissions** tab of the lambda function
- Change the Lambda Role Policy permission
    - Under **Execution role**, click on the hyperlink under **Role name**
    - On the **Permissions** tab, find the policy name and click on it
    - Choose **Edit policy**, and choose the **JSON** tab
    - Append the following to the Statement section of the policy to allow publishing to AWS IoT.

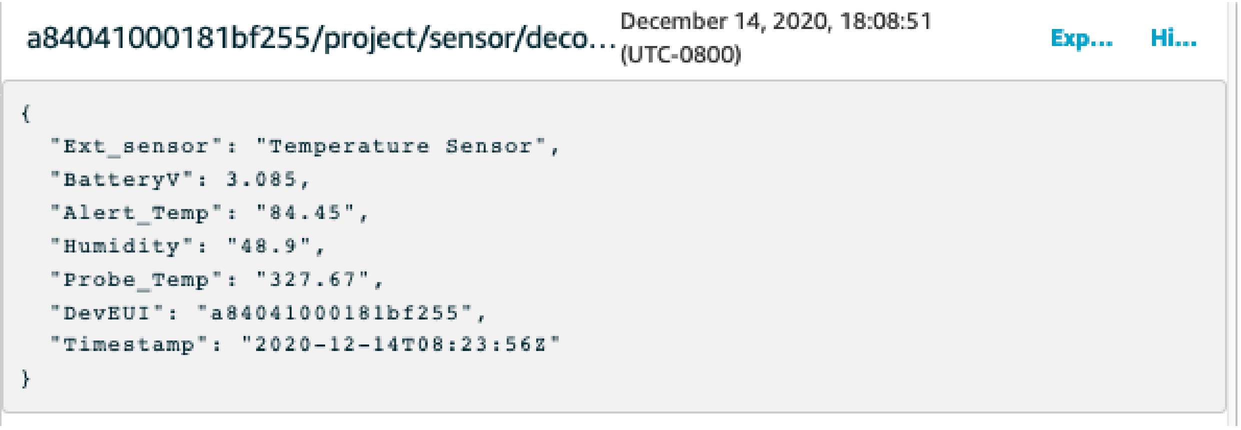
```
,
{
"Effect": "Allow",
"Action": [
"iot:Publish"
],
"Resource": [
"*"
]
}
```

- Choose **Review Policy**, then **Save changes**

- Create a test event that will allow you to test the functionality of the lambda function.
  - In the drop-down for *Select a test event*, choose **Configure test events**
  - Enter a name for the test event under **Event name**
  - Paste the following sample payload in the area under Event name:

```
{
    "MessageId": "55d122ab-6355-2233-9874-ff47c5222108",
    "WirelessDeviceId": "65d128ab-90dd-4668-9556-fe47c589610b",
    "PayloadData": "zA0LYgHpAX//f/8=",
    "WirelessMetadata":
    {
        "LoRaWAN":
        {
            "DevEui": "a84041000181bf255",
            "FPort": 2,
            "DataRate": 0,
            "Frequency": 904500000,
            "Gateways": [
             {
                    "GatewayEui": "80029cffXXXXXXXX",
                    "Snr": 12.25,
                    "Rssi": -47
             }
             ],
            "Timestamp": "2020-12-14T08:23:56Z",
        }
    }
}
```

- Choose **Create** to save the event
- Navigate to the AWS IoT console, choose **Test** on the navigation pane, and select **MQTT client**.
- Configure the MQTT client to subscribe to “#” (all topics)
- Click on **Test** in the Lambda function page to generate the test event you just created
- Verify the published data in the AWS IoT Core MQTT Test client
  - Open another window. Goto AWS IoT Console, select Test, under Subscription Topic, enter # and select to Subscribe to topic
  - The output should look similar to this:


a84041000181bf255/project/sensor/deco...
December 14, 2020, 18:08:51 (UTC-0800)
Exp...
Hi...

```
{
  "Ext_sensor": "Temperature Sensor",
  "BatteryV": 3.085,
  "Alert_Temp": "84.45",
  "Humidity": "48.9",
  "Probe_Temp": "327.67",
  "DevEUI": "a84041000181bf255",
  "Timestamp": "2020-12-14T08:23:56Z"
}
```

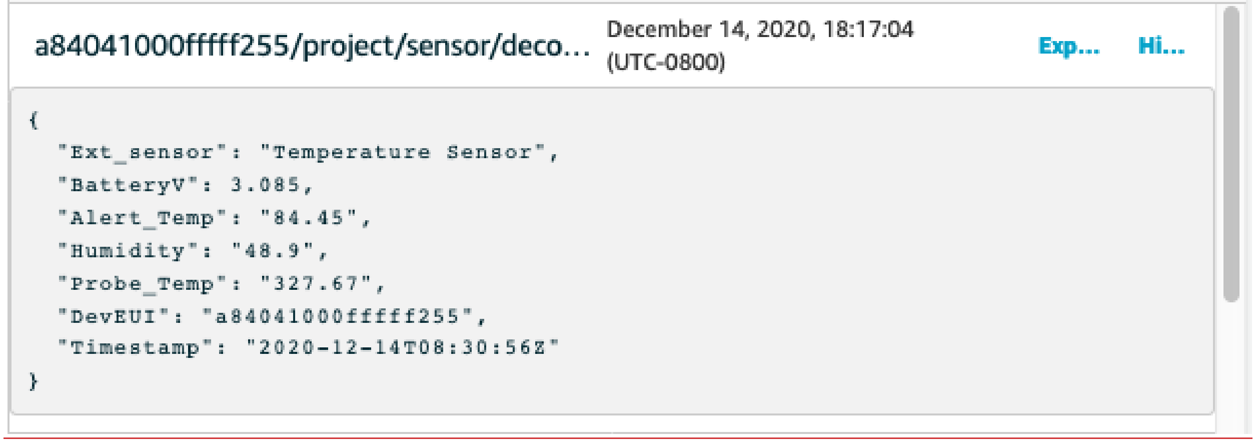
7.2 Create the Destination rule

In this step, you create the IoT rule that forwards the device payload to your application. This rule is associated with the destination created earlier in Set up a Destination for device traffic.

- Navigate to the AWS IoT console.
- In the navigation pane, choose **Act**. Then, choose **Rules**.
- On the Rules page, choose **Create**.
- On the **Create a rule** page, for **Name**, enter *LoRaWANRouting*. For **Description**, enter a description of your choice. Note the name of your rule. The information will be needed when you provision devices to run on AWS IoT Core for LoRaWAN.
- Leave the default Rule query statement: 'SELECT * FROM 'iot/topic' unchanged. This query has no effect at this time, as traffic is currently forwarded to the rules engine based on the destination.
- Under **Set one or more actions** choose Add action.
- On the Select an action page, choose **Republish a message to an AWS IoT topic**. Scroll down and choose **Configure action**.
- On the Configure action page, for **Topic**, enter *project/sensor/decoded*. The AWS IoT Rules Engine will forward messages to this topic.
- Under **Choose or create a role to grant AWS IoT access to perform this action**, choose **Create Role**.
- For **Name**, enter a name of your choice.
- Choose **Create role** to complete the role creation. You will see a "Policy Attached" tag next to the role name, indicating that the Rules Engine has been given permission to execute the action.
- Choose **Add action**.
- Add one more action to invoke the Lambda function. Under **Set one or more actions** choose **Add action**.
- Choose **Send a message to a Lambda function**
- Choose **Configure action**
- Select the *sailboatdecoder* lambda function created earlier and choose **Add action**
- Then, choose **Create rule**.
- A "Success" message will be displayed at the top of the panel, and the destination has a rule bound to it.

You can now check that the decoded data is received and republished by AWS by triggering a condition or event on the device itself.

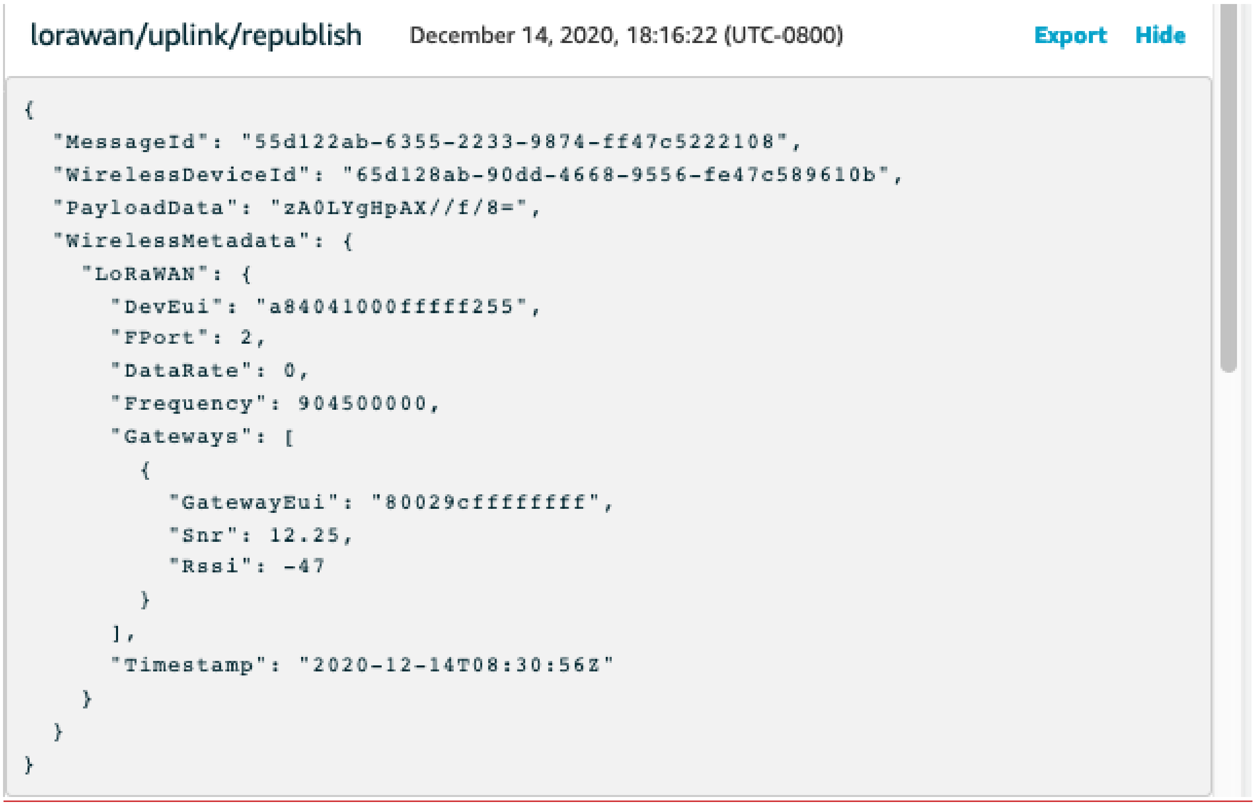
1. Go to the AWS IoT console. In the navigation pane, select **Test**, and choose **MQTT client**.
2. Subscribe to the wildcard topic '#" to receive messages from all topics
3. You should see traffic similar to that shown below.



a84041000fffff255/project/sensor/deco...

December 14, 2020, 18:17:04 (UTC-0800)

```
{
  "Ext_sensor": "Temperature Sensor",
  "BatteryV": 3.085,
  "Alert_Temp": "84.45",
  "Humidity": "48.9",
  "Probe_Temp": "327.67",
  "DevEUI": "a84041000ffff255",
  "Timestamp": "2020-12-14T08:30:56Z"
}
```



```
{
  "MessageId": "55d122ab-6355-2233-9874-ff47c5222108",
  "WirelessDeviceId": "65d128ab-90dd-4668-9556-fe47c589610b",
  "PayloadData": "zA0LYgHpAX//f/8=",
  "WirelessMetadata": {
    "LoRaWAN": {
      "DevEui": "a84041000ffff255",
      "FPort": 2,
      "DataRate": 0,
      "Frequency": 904500000,
      "Gateways": [
        {
          "GatewayEui": "80029cffffffff",
          "Snr": 12.25,
          "Rssi": -47
        }
      ],
      "Timestamp": "2020-12-14T08:30:56Z"
    }
  }
}
```

7.3 Configuring Amazon SNS

We will use the Amazon Simple Notification Service to send text messages (SMS) when certain conditions are met.

- Go to the Amazon SNS console.
- Click on the menu in the left corner to open the navigation pane.
- Select **Text Messaging (SMS)** and choose **Publish text message**.
- Under **Message type**, select **Promotional.**
- Enter your phone number (phone number that will receive text alerts)
- Enter “Test message” for the **Message** and choose **Publish message**.
- If the phone number you entered is valid, you will receive a text message and your phone number will be confirmed.
- Create an Amazon SNS Topic as follows:
  - In the navigation pane, choose **Topics**
  - Select **Create topic**
  - Under **Details**, select **Standard**
  - Enter a name of your choice. Here we will use “*text_topic*”.
  - Choose **Create topic**
- Create a subscription for this topic:
  - In the page for the newly created *text_topic*, choose the **Subscriptions** tab
  - Choose **Create subscription**
  - Select **Protocol** as *SMS* from the drop-down
  - Under **Endpoint**, enter the previously validated phone number to receive the SMS alerts
  - Choose **Create subscription**. You should see a *“Subscription to text_topic created successfully”* message.

7.3.1 Add a rule for Amazon SNS notification

Now add a new rule to send an Amazon SNS notification when certain conditions are met in a decoded message.

- Navigate to the AWS IoT console.
- In the navigation pane, choose **Act**. Then, choose **Rules**.
- On the Rules page, choose **Create**
- Enter the **Name** as *text_alert*, and provide an appropriate **Description**
- Under **Rule query statement**, enter the following query:
```
SELECT DevEUI as device_id, "Temperature exceeded 80" as message,
Alert_Temp as temp, Humidity as humidity, Timestamp as time FROM
'project/sensor/decoded' where Alert_Temp > 80
```
- Choose **Add action**
- Choose **Send a message as an SNS push notification**
- Choose **Configure action**
- Under **SNS target**, select *text_topic* from the drop-down
- Select *RAW* under **Message format**
- Under **Choose or create a role to grant AWS IoT access to perform this action**, choose **Create role.**
- Enter a name for the role and choose **Add action**
- Choose **Create rule**. You should see a “Success” message, indicating that the rule has been created.

7.4 IoT Analytics

We will use IoT Analytics to visually display data via graphs if there is a need in the future to do further analysis.

7.4.1 Create an IoT Analytics Rule

First create a rule

- Navigate to the AWS IoT console.
- In the navigation pane, choose **Act**. Then, choose **Rules**.
- On the Rules page, choose **Create**
- Enter the **Name** as *Visualize*, and provide an appropriate **Description**
- Under **Rule query statement**, enter the following query:
```
SELECT * FROM 'project/sensor/decoded'
```
- Choose **Add action**
- Select **Send a message to IoT Analytics**
- Choose **Configure Action**
- Choose **Quick Create IoT Analytics Resources**
- Under **Resource Prefix**, enter an appropriate prefix for your resources, such as *LoRa*
- Choose **Quick Create**
- Once the **Quick Create Finished** message is displayed, choose **Add action**.
- Choose **Create rule**. You should see a Success message, indicating that the rule has been created.

7.4.2 Configure AWS IoT Analytics

Set up AWS IoT Analytics as follows:

- Go to the AWS IoT Analytics console.
- In the navigation panel, choose **Data sets**
- Select the data set that was generated by the Quick Create in Create an IoT Analytics Rule
- In the **Details** section, **Edit** the **SQL query**.
- Replace the query with:

```
select Alert_Temp as temp, Humidity as humidity, DevEUI as device_id, Timestamp
as time from LoRa_datastore
```

- Under **Schedule**, choose **Add schedule**
- Under **Frequency**, choose **Every 1 minute**, and choose **Save**

7.4.3 Configure Amazon QuickSight

Amazon QuickSight lets you easily create and publish interactive BI dashboards that include Machine Learning-powered insights.

- Go to AWS Management console.
- From the management console, enter "QuickSight" in the "*Search for services, features..*" search box.
- Click on **QuickSight** in the search results
- If you haven't signed up for the service before, go ahead and sign up, as there is a free trial period.
- Select the **Standard** Edition, and choose **Continue**
- Enter a unique name in the field **QuickSight account name**
- Fill in the **Notification email address**
- Review the other checkbox options and change them as necessary. The **AWS IoT Analytics** option must be selected.
- Choose **Finish.** You will see a confirmation message.
- Choose **Go to Amazon QuickSight**
- Select **Datasets**
- Select **New dataset**
- Select **AWS IoT Analytics**
- Under **Select an AWS IoT Analytics data set to import**, choose the data set created in Create an IoT Analytics Rule
- Choose **Create data source**, and then choose **Visualize**
- Select dataset created, then select **Refresh** or **Schedule Refresh** for periodic refresh of dataset.

7.5 Testing your "Hello World" Application

Using your device, create a condition to generate an event such as a high temperature condition. If the temperature is above the configured threshold then you will receive a text alert on your phone. This alert will include key parameters about the alert.

You can also visualize the data set as follows:

- Go to the AWS IoT Analytics console
- Choose **Data sets**
- Select the dataset created earlier
- Select **Content**. and ensure there are at least few uplink entries available in the data set.
- Go to the QuickSight console
- Choose **New analysis**
- Choose the dataset created in Create an IoT Analytics Rule
- Select time on the X-axis, Value as temp (Average) and Color as device_id to see a chart of your dataset.

8 Debugging

The gateway's logs can be accessed in the web frontend under **Status->System Log** and **Status->Kernel Log.**

The linux console can be accessed using **ssh root@<ipaddress>** and the same password as the web frontend.

9 Troubleshooting

- Make sure the gateway has power and is connected to the internet.
- If LTE connection is used, check if your SIM card has enough data volume left
- Verify that the GatewayEUI is correct (see Section 4.3)
- Verify that the endpoints (URI and port) and certificates and region are correct
- try unplugging and re-plugging power
- Try a different network and check your firewall settings
- Update to the newest firmware: https://docs.miromico.ch/miroEdge/update.html (the gateway needs to be reconfigured after the update)