LORIX One
Getting Started Guide for AWS IoT Core for LoRaWAN

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)

Revision	*Date*	*Author*	*Changes*
v1.0	*08.08.22*	*Christian Müller*	*Initial redaction of the Getting started guide*
v1.1	*29.08.22*	*Christian Müller*	*Complete and improve section 7.1 (lambda function)*

2 Overview

The LORIX One is compact and robust LoRaWAN® base station (gateway), ideal for all kind of deployments, simple or complex.

Its convenient size makes it easy to install anywhere, and its thoughtful design and quality materials make it reliable and resistant to the harshest environments.

The embedded software is reliable, secure and intuitive to use via its modern and clear web interface.

3 Hardware Description

3.1 DataSheet

https://iot.wifx.net/en/products/lorix-one/#specifications.

3.2 Standard Kit Contents

- LORIX One base station (including cap and sealing grommet)
- Antenna (indoor/outdoor 3dBi/outdoor 5dBi depending on reference)
- Passive PoE injector
- 24V power supply for PoE injector (including EU/US/UK/AU power plug adapter)
- 2x UV resistant cable tie
- Quick start guide

3.3 User Provided items

- RJ45 Ethernet cable
- Network providing equipment (router, switch, hub)
- A power source
- A terminal (computer, smartphone, tablet) running a modern web browser (or a terminal emulator with SSH)
- If you cannot access the gateway through the network, you will need a USB to mini-USB cable for debugging.

3.4 3rd Party purchasable items

- Antenna-gateway surge protection : JIASIDA SP3000
- Gateway-switch surge protection : UBIQUITI ETH-SP-G2
- Outdoor PoE splitter : Ligowave DLB-POE-SPLITTER-03

3.5 Additional Hardware References

You can get more information about the regional variants on https://iot.wifx.net/en/products/lorix-one/#specifications.

4 Setup AWS

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. Onboard your Gateway (see section Add the Gateway to AWS IoT)
2. Onboard your 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 Add the Gateway to AWS IoT

4.2.1 Preparation

Refer to the online guide for steps required prior to on-boarding your gateway.

You need LORIX OS 1.0 or greater, the legacy firmware is not supported. Please see https://iot.wifx.net/docs/lorix-one/firmware-versions for details.

4.2.2 Frequency Band selection and Role setup

The following frequency bands are supported:

8XX version:

- EU868
- IN865
- RU864

9XX version:

- AS920
- AS923
- AU915 (Subband 1-8)
- US915 (Subband 1-8)

Refer to the online guide for information on selecting an appropriate frequency band.

Follow the instructions in the section **Add an IAM role to allow the Configuration** and **Update Server (CUPS) to manage gateway credentials** in the online guide**.**

4.2.3 Add the LoRaWAN Gateway

To register the Gateway with AWS IoT Core for LoRaWAN, follow the steps in this online guide under the section **Add a gateway using the console**.

Please see the Wifx IoT documentation for details regarding the Gateway EUI.

4.3 Add a LoRaWAN Device to AWS IoT

4.3.1 Preparation

Refer to the instructions in the section **Before onboarding your wireless device** in the online guide.

Then follow the instructions in the section **Add your wireless device to AWS IoT Core for LoRaWAN** here.

4.3.2 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. For more details, refer to the section **Add profiles to AWS IoT Core for LoRaWAN** in the online guide.

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

4.3.3 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, refer to the online guide (sections titled **Add a destination using the console** and **Create an IAM role for your destinations**). Also refer to **Create rules to process LoRaWAN device messages** in the online guide.

5 Set up the Gateway

5.1 Set up Gateway hardware

Mount and connect the gateway following the online instructions:

- Mount the gateway on a pole/wall
- Power up the gateway

5.2 Set up Gateway Software

Setup the gateway by following the “Setup” guide of the LORIX OS User guide at https://iot.wifx.net/docs/lorix-os/latest/lorix-one/user-s-guide.

5.3 Additional Software References

5.3.1 Gateway operating modes and reset

To perform a factory reset, please see : https://iot.wifx.net/docs/lorix-one/latest/lorix-os/hardware/operating-modes-and-reset.

5.3.2 OTA system updates

To update the system to the latest version, please check https://iot.wifx.net/docs/lorix-os/latest/lorix-one/user-s-guide/system/system-upgrade.

5.4 Configure the Gateway device

Follow the online guide to configure the Basic Station. Use the credentials obtained in 4.2 to configure the Basic Station.

5.5 Connect the Gateway and verify the connection status

Follow the instructions in the online guide to connect your gateway to AWS IoT Core for LoRaWAN.

To verify the connection status, refer to the instructions in the section **Check gateway connection status using the console.**

6 Add End Device(s)

6.1 Connect the device and verify the connection status

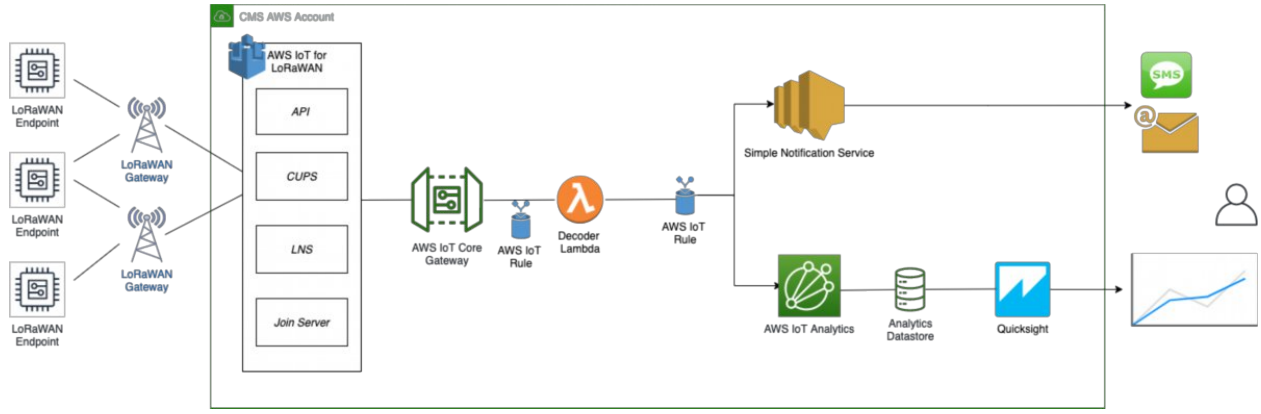
Follow the instructions in the online guide to connect your device to AWS IoT Core for LoRaWAN.

To verify the connection status, refer to the instructions in the section **Check device connection status using the console.** You can also **View format of uplink messages sent from LoRaWAN devices**.

7 Verifying Operation – a "Hello World" example

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 Python 3.9*. from the drop-down under **Runtime**.
- Click on **Create function**.
- Navigate to *https://github.com/aws-samples/aws-iot-core-lorawan/blob/main/transform_binary_payload/src-iotrule-transformation/app.py* and copy the code for the lambda function.
- Under **Function code**, paste the copied code into the editor under the lambda_function.py tab.
- Choose the payload decoder corresponding to your device from the VALID_PAYLOAD_DECODER_NAMES list (if needed, create your own payload decoder as instructed in the file) and copy the decode function file from https://github.com/aws-samples/aws-iot-core-lorawan/tree/main/transform_binary_payload/src-payload-decoders/python to the function folder.
- Once done, 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": [
        "*"
    ]
}
```

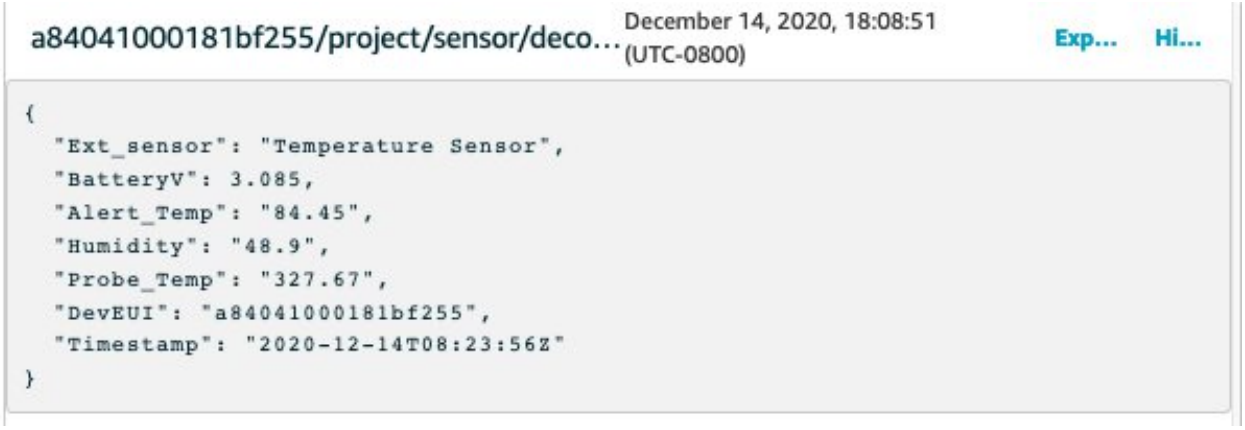
- Choose **Review Policy**, then **Save changes**

***NOTE** – The examples in this document are intended only for dev environments. All devices in your production 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.*

- 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:



7.2 Create the Destination rule

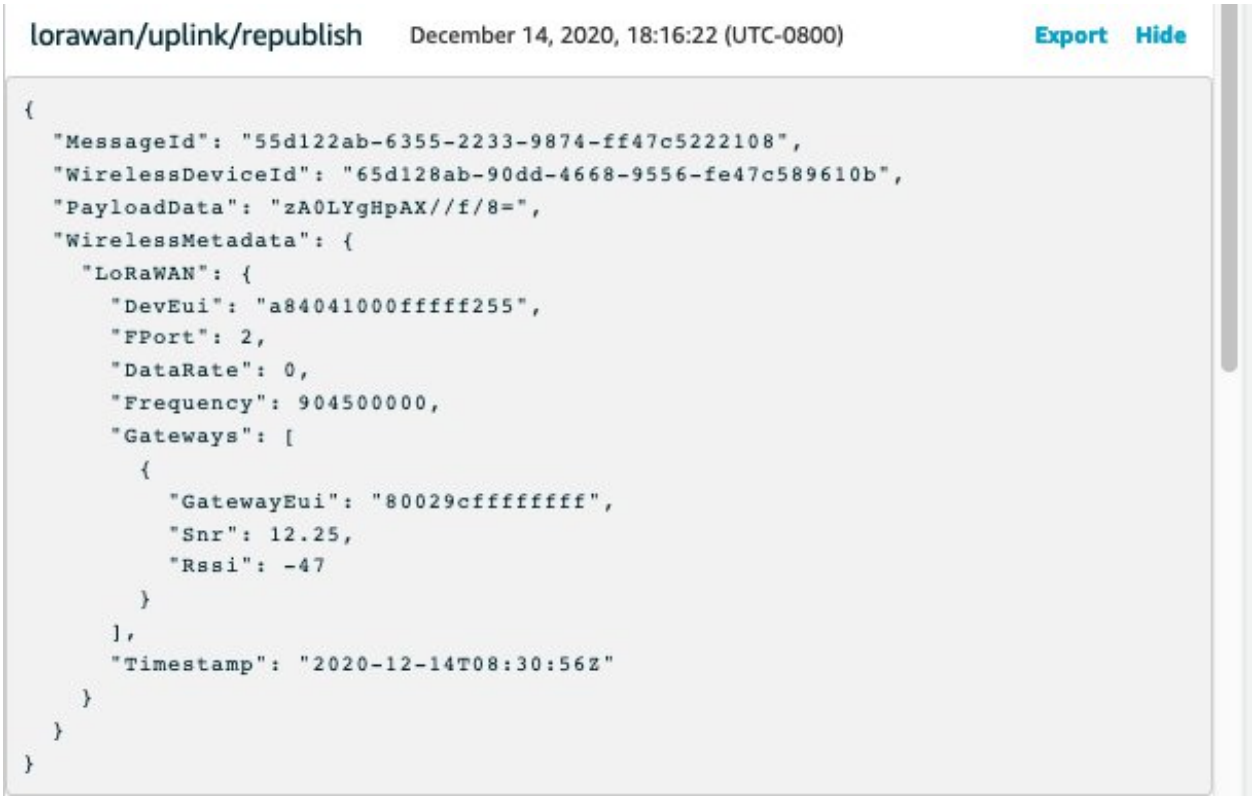
In this step, you create the IoT rule that forwards the device payload to your application. This rule is associated with the destination created earlier in Set up a Destination for device traffic.

- 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 Test client**.
2. Subscribe to the wildcard topic '#" to receive messages from all topics

3. You should see traffic similar to that shown below.



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

For debugging, please check the gateway Logs.

9 Troubleshooting

For general troubleshooting related to the gateway and LORIX OS, please check:

- https://iot.wifx.net/docs/lorix-os/latest/lorix-one/troubleshooting
- https://iot.wifx.net/docs/lorix-one/latest/lorix-os/troubleshooting

For troubleshooting of the Basic Station, please check:

- https://iot.wifx.net/docs/lorix-os/latest/lorix-one/user-s-guide/lorawan/packet-forwarders/basic-station/troubleshooting

10 OTA Updates

OTA updates of the system are done by LORIX OS directly, and not using the Basic Station OTA update system. Please refer to 5.3.2 for details.