Dragino – DLOS8, LG308, LPS8, LIG16 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)

V1.0 Date 2021-May-11 Release

2 Overview

This document shows how to set up Dragino LoRaWAN gateway to work with AWS IoT Core for LoRaWAN.

3 Hardware Description

3.1 DataSheet

Support Hardware are:
DLOS8:
https://www.dragino.com/downloads/index.php?dir=LoRa_Gateway/DLOS8/&file=Datasheet_DLOS8_LoRaWAN_Gateway.pdf
LG308: https://www.dragino.com/downloads/index.php?dir=LoRa_Gateway/LG308-LG301/&file=Datasheet_LG308_LoRaWAN_Gateway.pdf
LPS8:
https://www.dragino.com/downloads/index.php?dir=LoRa_Gateway/LPS8/&file=Datasheet_LPS8_LoRaWAN%20Pico%20Station.pdf
LIG16:
https://www.dragino.com/downloads/index.php?dir=LoRa_Gateway/LIG16/&file=Datasheet_LIG16_LoRaWAN%20Indoor%20Gateway.pdf

Above hardware has the same method to connect to AWS IoT Core for LoRaWAN*. Suggested firmware version > lgw--build-v5.4.1615882321-20210316-1613*

3.2 Standard Kit Contents

The gateway mentioned above already include power adapter, User still need a RJ45 cable to connect and configure it.
Above gateway can be found on Dragino Official Website: https://www.dragino.com

3.3 User Provided items

User needs a RJ45 cable and PC to configure the gateway.

3.4 3rd Party purchasable items

For gateway connection, above items are enough.

3.5 Additional Hardware References

Additional hardware from Dragino can be found on https://www.dragino.com

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 (Gateway EUI) or serial number. This is used to establish the connection between the LNS and the gateway. Consult the documentation for your gateway to locate this value.

User can get the Gateway ID from Dragino Gateway Web UI:

- The gateway with the firmware version higher than lgw--build-v5.4.1615882321-20210316-1613 is required. (Click here to check gateway version).

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** 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). The cert.pem and private.key
    - *Add a note if your gateway requires files of a specific name or non-pem format.*
- 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.
- The xxxxx.cert.pem, xxxxxx.private.key and cups.trust will be needed to upload to Gateway in set up gateway section.

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 thedestination
- 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

Dragino Gateway models LPS8,LIG16,DLOS8, LG308 all support AWS IoT Core for LoRaWAN, they have the same con figuration method to connect to AWS IoT Core for LoRaWAN. The example below uses LIG16 as reference.

5.1.1 Choose the power supply

- For LIG16, Choose a 5V2A USB adapter.
- For other models, use the power adapter shipped with the devices.

5.1.2 LED Indicators

- ➢ **Power LED** : This YELLOW LED will be solid on if the device is properly powered.
- ➢ **HEART LED** : This GREEN LED will be solid on if there is LoRaWAN connection.
- ➢ **SYS LED** :This LED will show different colors on different state:
- ✓ ON: device have Internet connection.
- ✓ BLINKING: a) Device has internet connection but no LoRaWAN Connection. or b)
- Device is in booting stage, in this stage, it will be BLINKING for several seconds.
- ✓ OFF: device doesn't have Internet connection.
- ➢ **TRIANGLE LED** : No Function.
- ➢ **ETH LED** : This LED shows the ETH interface physical connection status.
- ➢ **WiFi LED** : This LED shows the WiFi interface connection status.
-

5.1.3 Access the Internet with DHCP IP from router

Connect the Gateway's WAN port to your router and Gateway can obtain an IP address from the router to have internet access. In the router's management portal, you should be able to find what IP address the router has assigned to the Gateway. You can use this IP to connect to the gateway.

5.1.4 Gateway Connect via WiFi

At the first boot of Dragino gateway, it will auto-generate an unsecure WiFi network called **dragino-xxxxxx**

Note: It has been password protected and the password is:

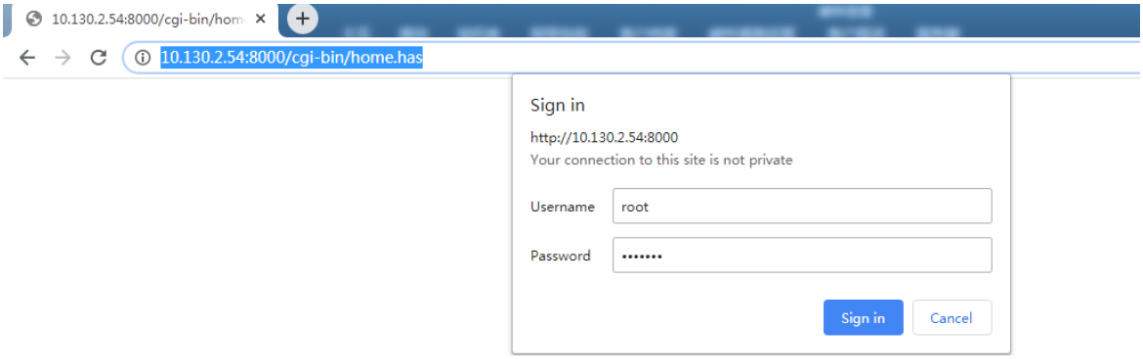
dragino+dragino

User can use the laptop to connect to this WiFi network. The laptop will get an IP address 10.130.1.xxx and the LG308 has the default IP **10.130.1.1**

5.1.5 Access Configure Web UI

Open a browser on the PC and type the LIG16 ip address (depends on your connect method)
http://10.130.1.1/ (Access via WiFi AP network)
or
http://IP_ADDRESS:8000 (If the IP is assigned by uplink router)
You will see the login interface of LIG16 as shown below.
The account details for Web Login are:

User Name: root
Password: dragino



5.2 Set up Gateway Software

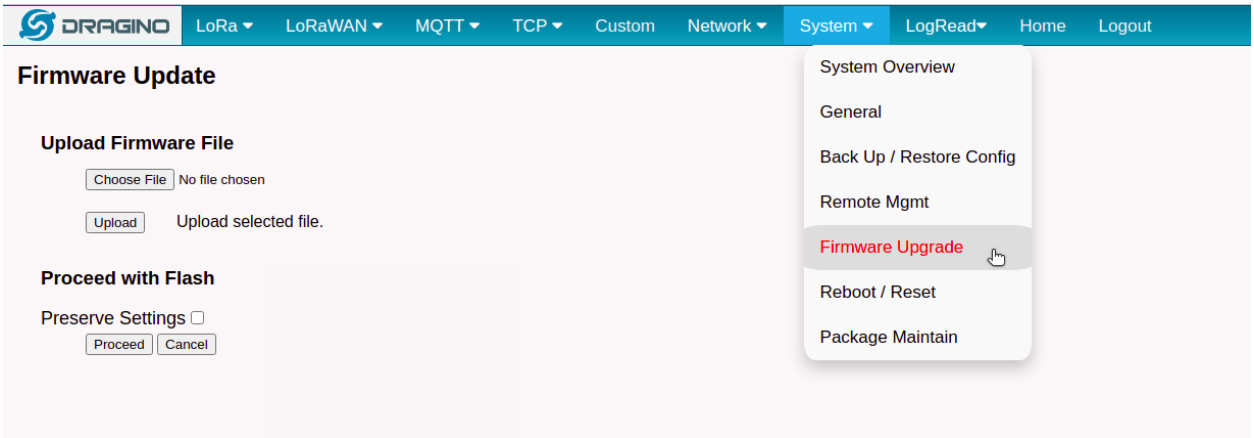
5.2.1 How to set up Gateway Software

The user only needs to set up the Amazon AWS IoT configuration UI at the gateway UI.
LoRaWAN --> Amazon AWS IoT.

Detail of set up, please see Section 5.4

5.2.2 Firmware upgrade for Gateway

Dragino gateway firmware versions >= lgw--build-v5.4.1615882321-20210316-1613 support AWS-IoT LoRaWAN Core.



5.3 Additional Software References

5.3.1 website

- Company Website: www.dragino.com
- Gateway User Manual:
  - ✓ LIG16: https://www.dragino.com/downloads/index.php?dir=LoRa_Gateway/LIG16/
  - ✓ LPS8: https://www.dragino.com/downloads/index.php?dir=LoRa_Gateway/LPS8/
  - ✓ LG308: https://www.dragino.com/downloads/index.php?dir=LoRa_Gateway/LG308-LG301/
  - ✓ DLOS8: https://www.dragino.com/downloads/index.php?dir=LoRa_Gateway/DLOS8/
- more detail and troubleshooting on how to set up the gateway with AWS-IoT:
  https://wiki.dragino.com/index.php?title=Notes_for_#Introduction

5.3.2 support

Support Email : support@dragino.com

5.4 Configure the Gateway

5.4.1 How to Upload a certificate to Gateway

The user needs to upload the certificate obtained by AWS to the gateway by accessing the gateway AWS-IoT UI. Below is the update page in gateway:

User need to:

a) Put the CUPS URI from AWS IoT Core From LoRaWAN to the CUPS URI field.
b) Make sure the Gateway ID is the same the Gateway EUI from AWS-IoT portal.
c) Upload the CUPS.trust file from AWS IoT Core for LoRaWAN to the Gateway
d) Upload Private Key from AWS IoT Core for LoRaWAN to the Gateway
e) Upload Cert Pem file from AWS IoT Core for LoRaWAN to the Gateway

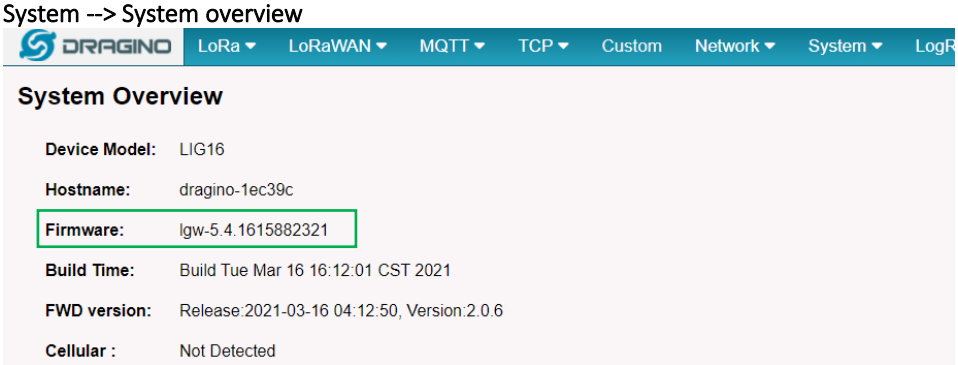
CUPS.trust / Private Key/ Cert Pem file can be obtained from AWS IoT console, refer section 4.3.2 Add the LoRaWAN Gateway

After upload the files and configure, user will be able to see below:

5.4.2 how to check firmware version

In System overview, user will see the Gateway version

System --> System overview



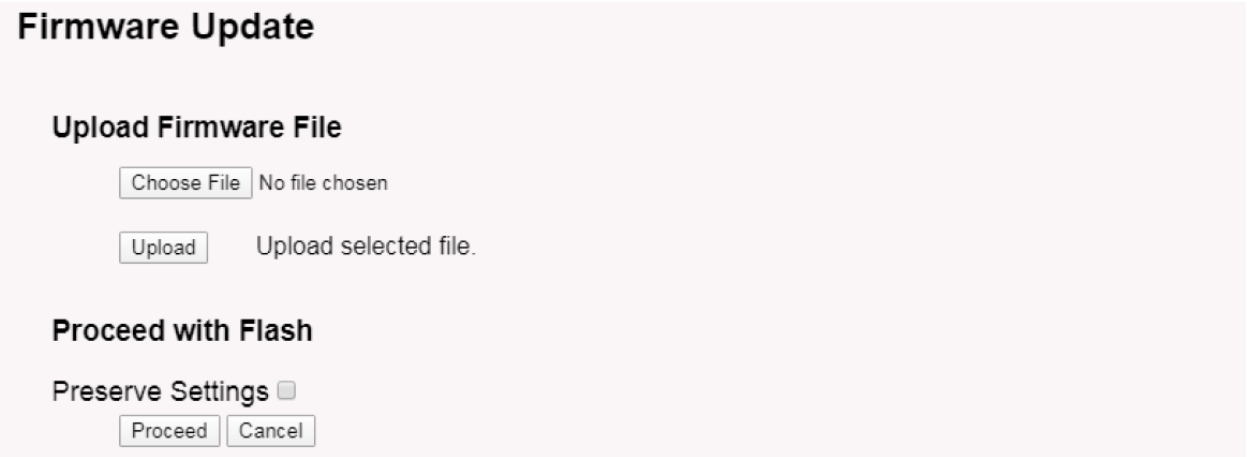
Suggested firmware version is lgw--build-v5.4.1615882321-20210316-1613 or later

5.4.3 Get the latest firmware

User can get the firmware from this link and update the firmware:
https://www.dragino.com/downloads/index.php?dir=LoRa_Gateway/LIG16/Firmware/Release/
The file named as dragino-lgw–xxxxx-squashfs-sysupgrade.bin is the upgrade Image.

5.4.4 How to upgrade

In gateway UI select System --> Firmware Upgrade



Select the required image and click Upload. The image will be uploaded to the device, and then click Process to upgr ade.

NOTE: You normally need to uncheck the Preserve Settings checkbox when doing an upgrade to ensure that there is no conflict between the old settings and the new firmware. The new firmware will start up with its default settings

6 Add End Device(s)

When user connect the gateway to AWS IoT Core for LoRaWAN, user just needs to add the end device to AWS IoT Core for LoRaWAN and start it, and the end device will start communicating with the gateway

6.1 How to add end device

Read 4.4 and add the device to AWS IoT Core for LoRaWAN.

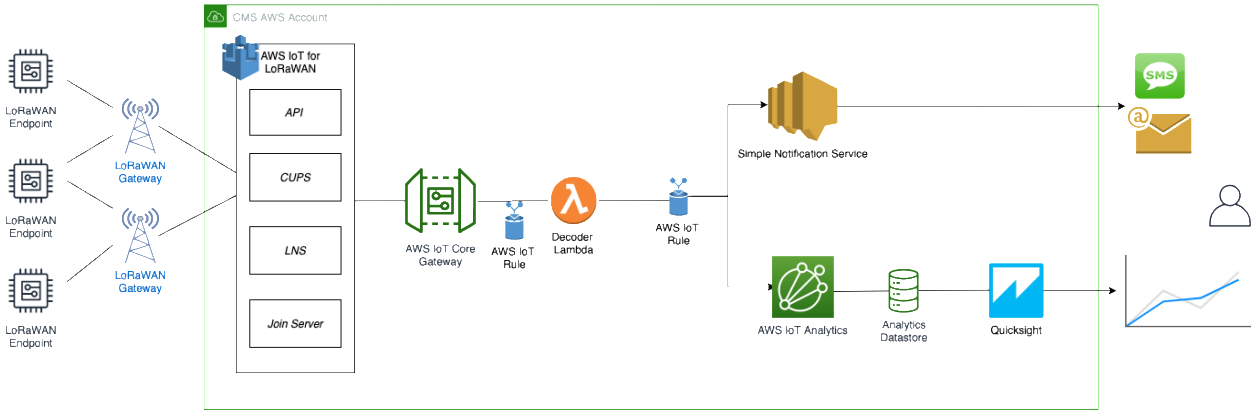
Or reference
https://wiki.dragino.com/index.php?title=Notes_for_AWS-IoT-Core#Add_wireless_device

7 Verifying Operation – a "Hello World" example

As an example, add Gateway-Dragino-LIG16 to connect to AWS IoT Core for LoRaWAN and add Dragino End Device-L HT65 to communicate with the gateway.

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

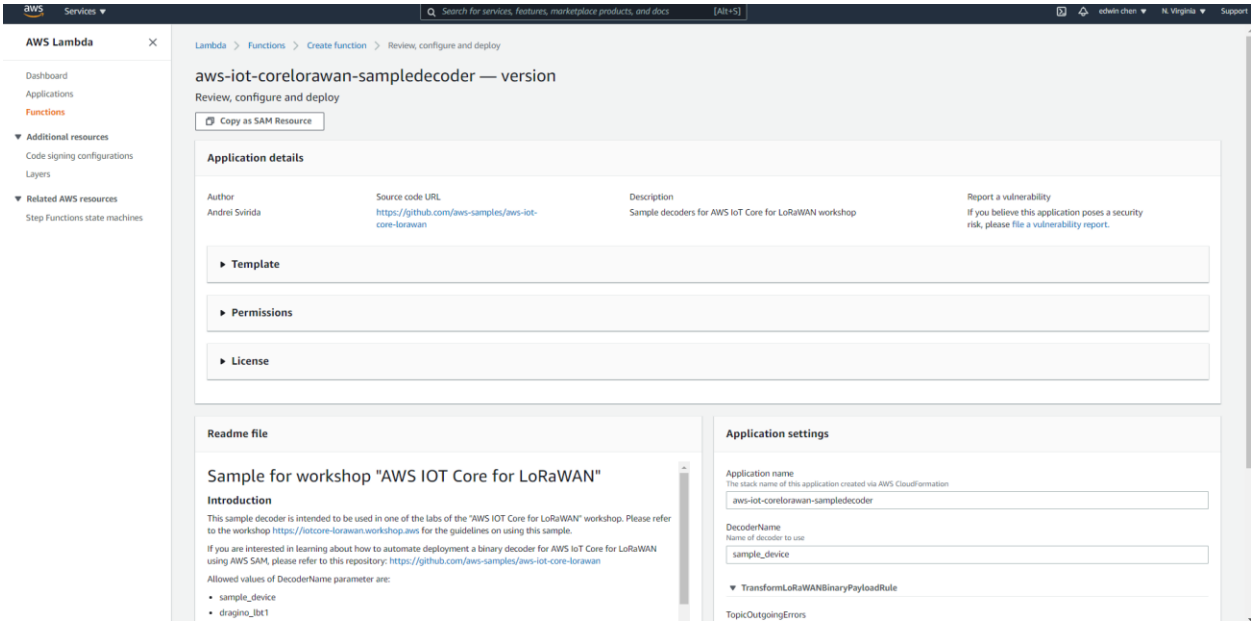
Create the lambda function to process device messages processed by the destination rule.

- Start deployment of a serverless application with AWS Lambda function and AWS IoT Rule
- Select a decoder
- Review deployment
- Create the test event

- provide PayloadData sample
- Run a test
- Note the AWS lambda function ARN
- Optional: review the source code of the binary decoder

Step 1: Start deployment of a serverless application with AWS Lambda function and AWS IoT Rule

Open AWS Lambda console by clicking on link https://console.aws.amazon.com/lambda/home?region=us-east-1#/create/app?applicationId=arn:aws:serverlessrepo:us-east-1:614797420359:applications/aws-iot-corelorawan-sampledecoder

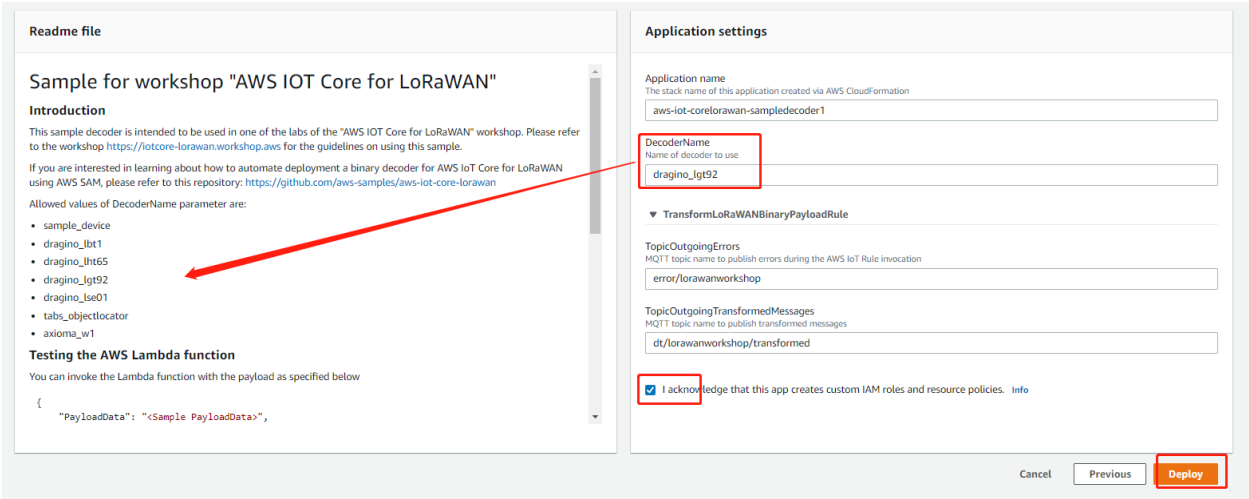


Step 2: Select a decoder

Please scroll down to the bottom of the page. Please provide the parameter DecoderName based on the following ta ble.

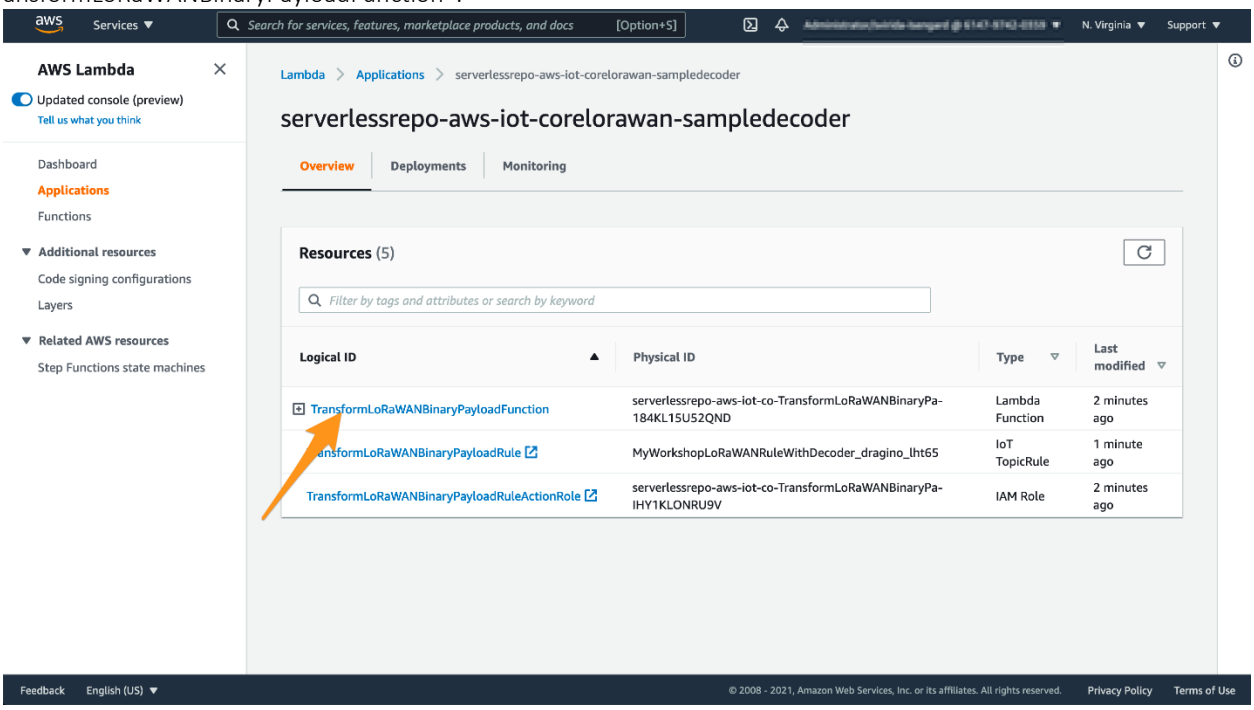
Device	Decoder name
LHT65	dragino_lht65
LBT1	dragino_lbt1
LSE01	dragino_lse01
LGT92	dragino_lgt92
LDS01	dragino_lds01

After that please check the box “I acknowledge that this app creates IAM roles and resource policies.” and click on “Deploy”.



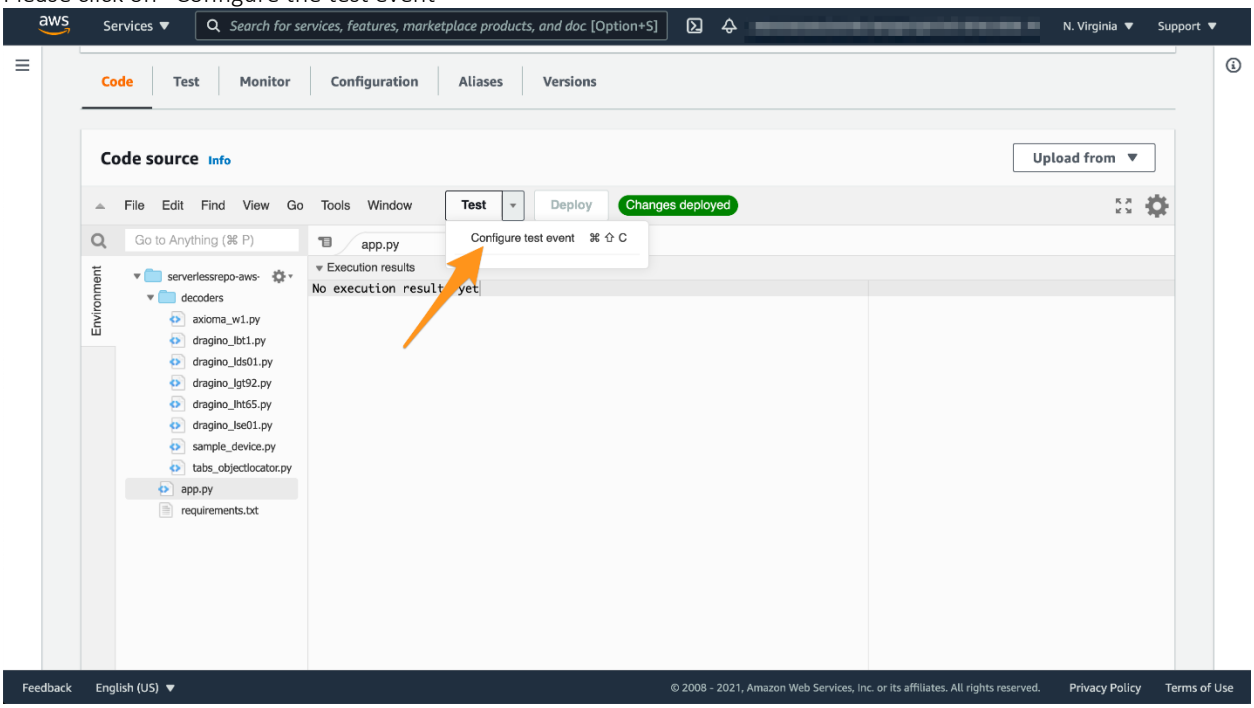
Step 3: Review deployment

Please wait few seconds for a successful deployment. After that please click on the name of the Lambda function “TransformLoRaWANBinaryPayloadFunction”.



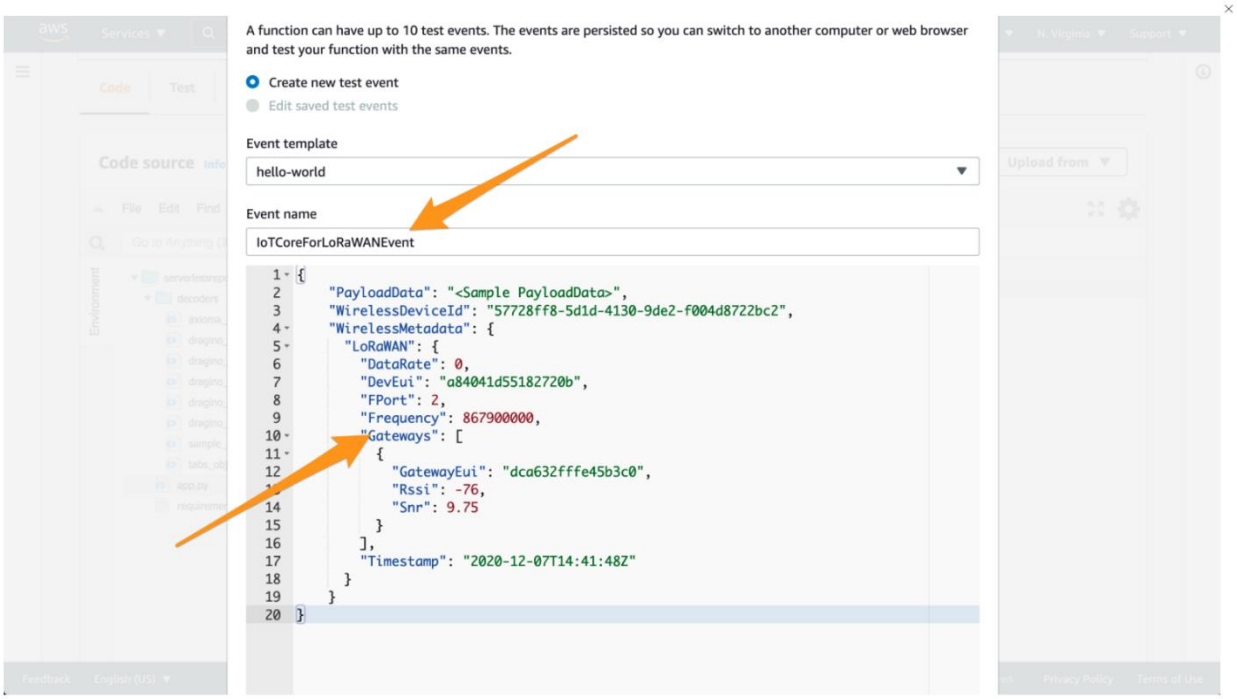
Step 4: Create the test event

Please click on “Configure the test event”



In the window that opens, please provide the event name e.g. IoTCoreForLoRaWANEvent. After that please paste th e following JSON content:

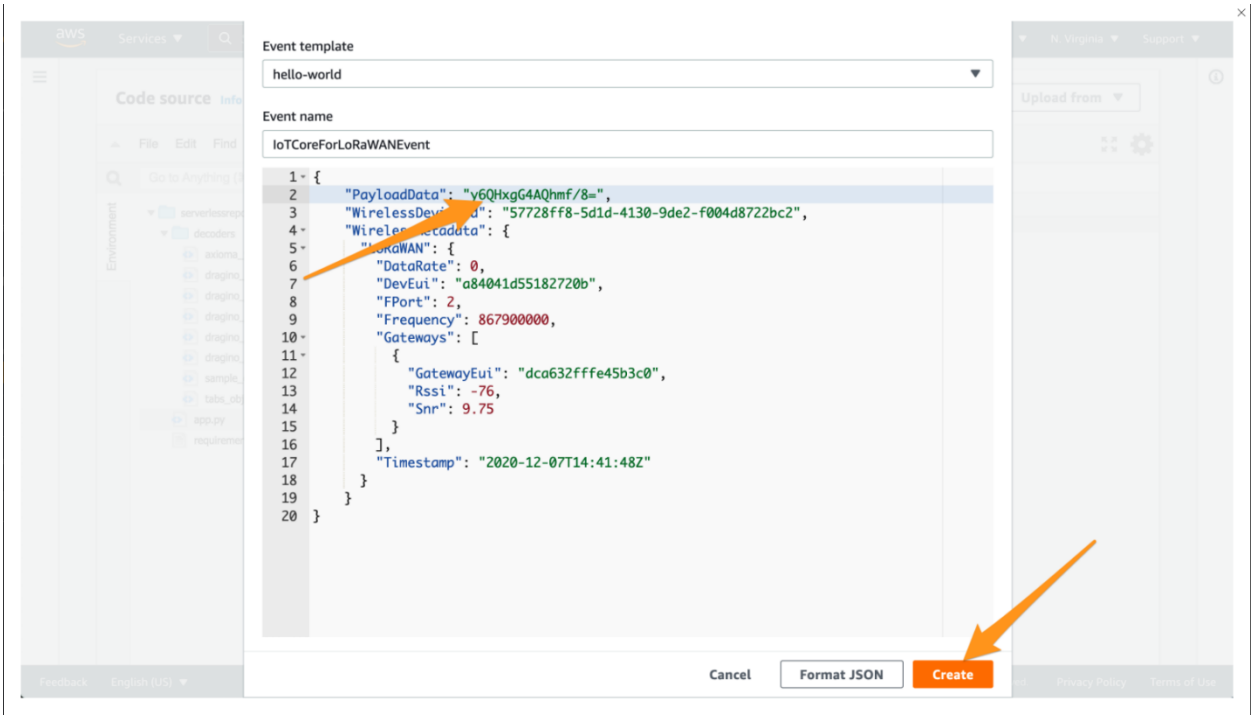

```
{
  "PayloadData": "<Sample PayloadData>",
  "WirelessDeviceId": "57728ff8-5d1d-4130-9de2-f004d8722bc2",
  "WirelessMetadata": {
    "LoRaWAN": {
      "DataRate": 0,
      "DevEui": "a84041d55182720b",
      "FPort": 2,
      "Frequency": 867900000,
      "Gateways": [
        {
          "GatewayEui": "dca632fffe45b3c0",
          "Rssi": -76,
          "Snr": 9.75
        }
      ],
      "Timestamp": "2020-12-07T14:41:48Z"
    }
  }
}
```



Step 5: Provide PayloadData sample

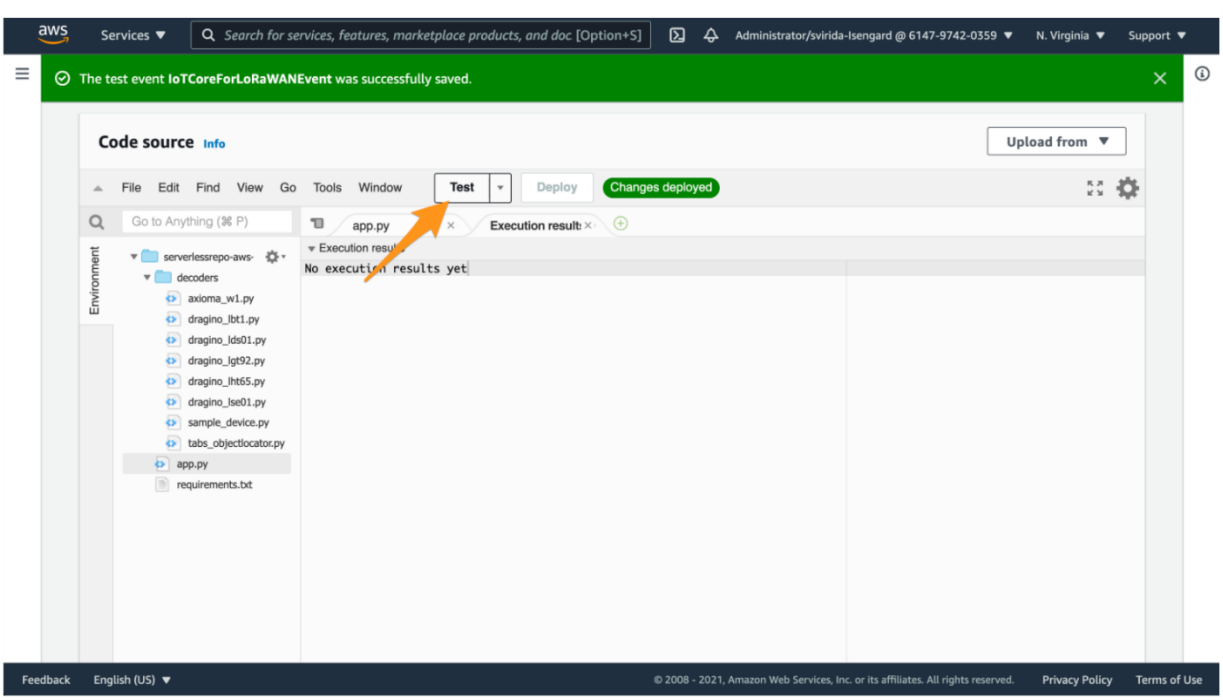
Please replace the string **<Sample PayloadData>** in the JSON document with a sample payload for the device you sel ected in step 2 according to this table. After that please click on “Create”.

Device name	Sample PayloadDate
LHT65	y6QHxgG4AQhmf/8=
LSE01	AuHtlACmawQPVGM=
LGT92	DSEAAAEVCMUGpAA=
LBT1	DxwAAAIDQUJCQ0NEREVFRkYwMjcxMjFGNkFDMy0wNTk=

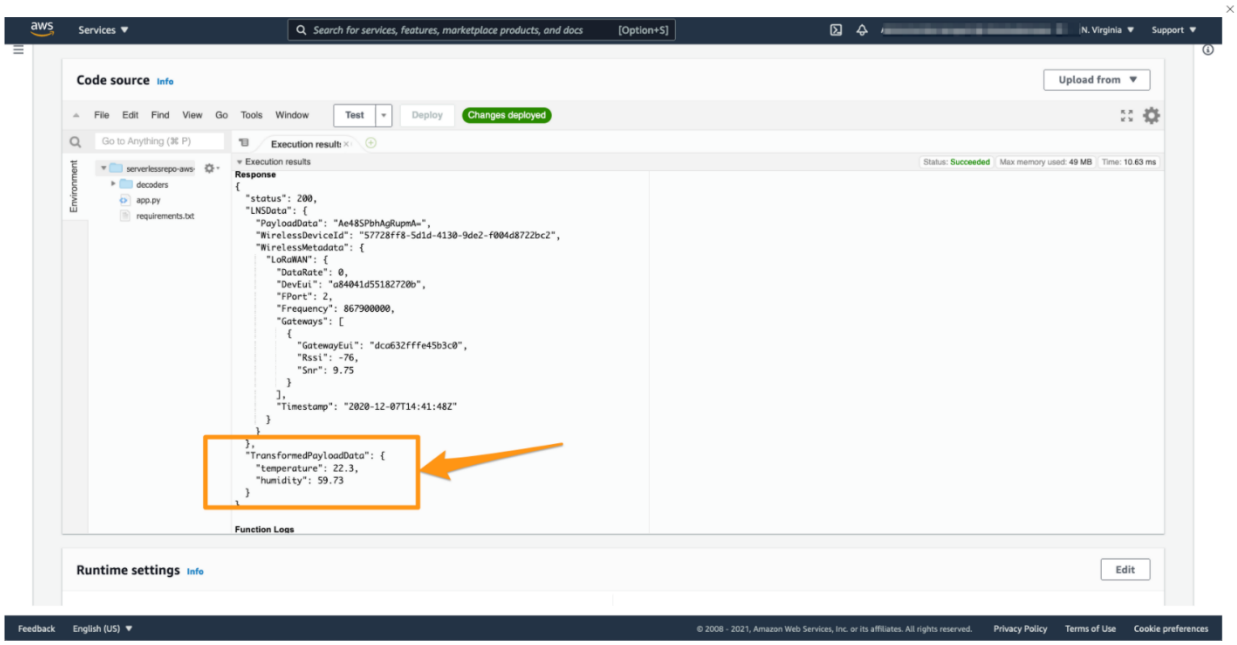


Step 6: Run a test

Click on "Test"

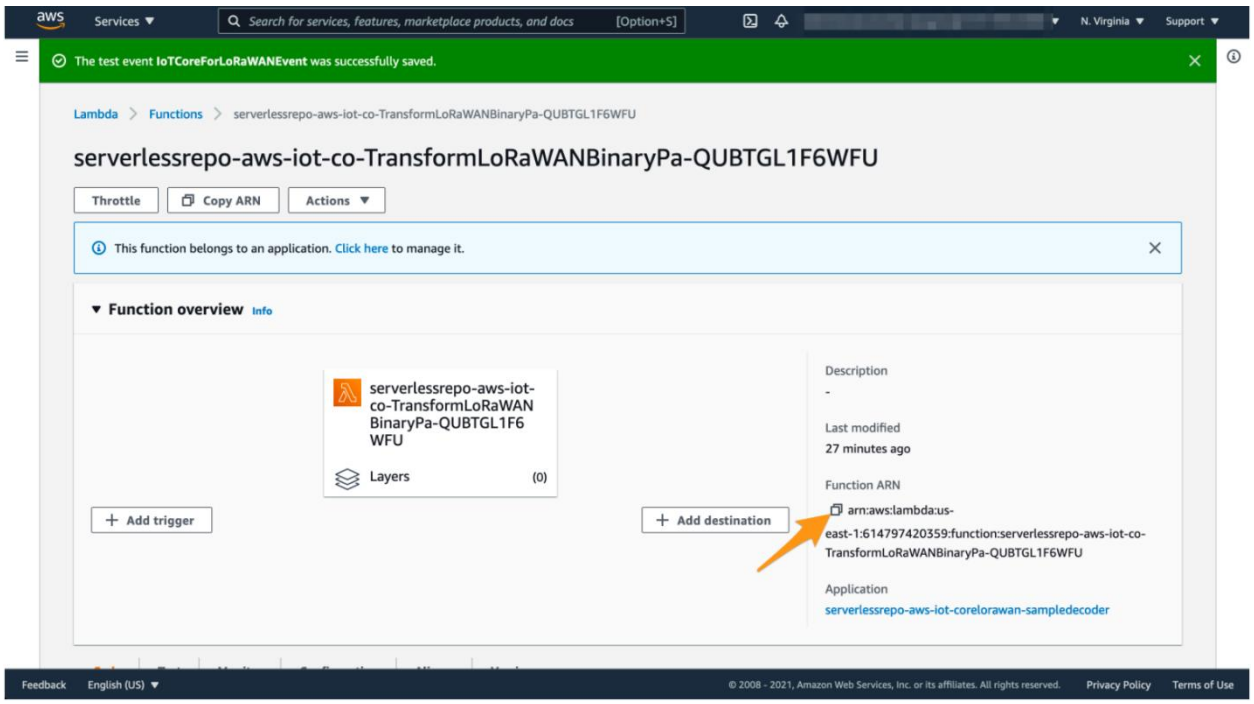


In the attribute TransformedPayloadData you can observe the result of binary decoding for the payload you specified in step 5. The output below assumes that you have selected “sample device” in step 2. It will contain other attributes if you have selected another decoder in step 2.



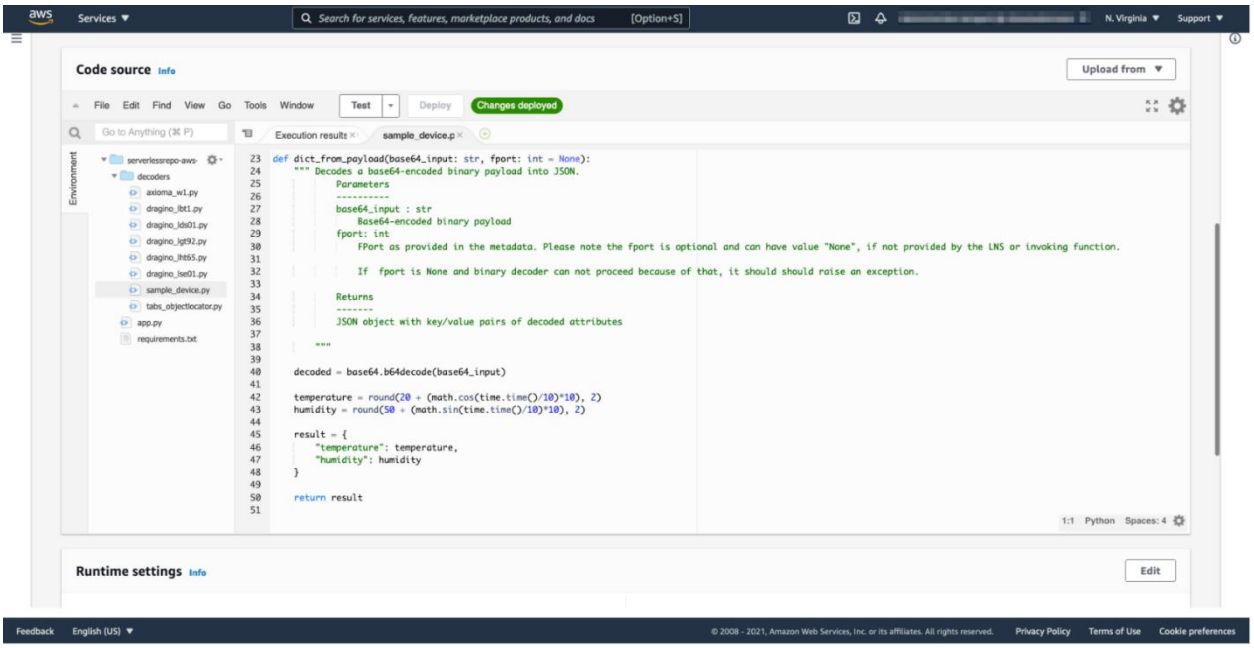
Step 7: Note the AWS lambda function ARN

As a preparation for the next step, please note the ARN of the deployed AWS Lambda function. We will need this AR N later to review the created AWS IoT Rule.



Step 8 : Optional: review the source code of the binary decoder

Though not required for the purpose of this workshop, feel invited to switch to the "Code" section of the AWS Lamb da function and inspect the Python source code. You will find the decoders for the individual devices in the **directory decoders.** The AWS Lambda function handler is in file **app.py**



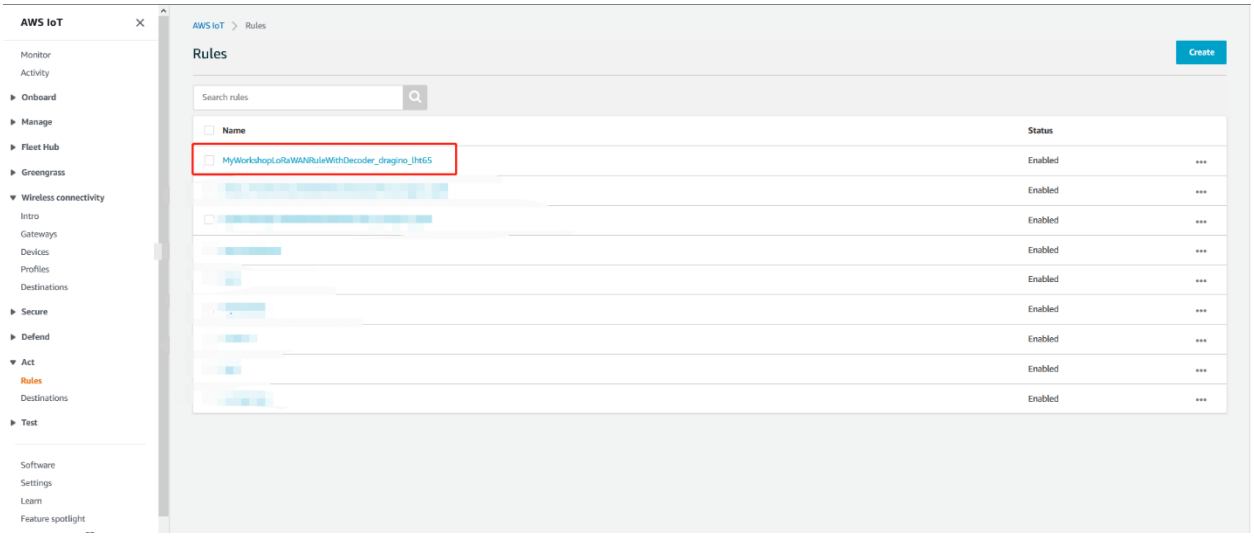
7.2 Update the Destination rule and get device's payload

In this step, you update 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.

- Find the IoT Rule
- Create a Destination with IoT Rule (MyWorkshopLoRaWANRuleWithDecoder_dragino_lht65)
- Update the destination to the device
- Check the payload
- Approach A with MQTT
- Approach B with Lambda

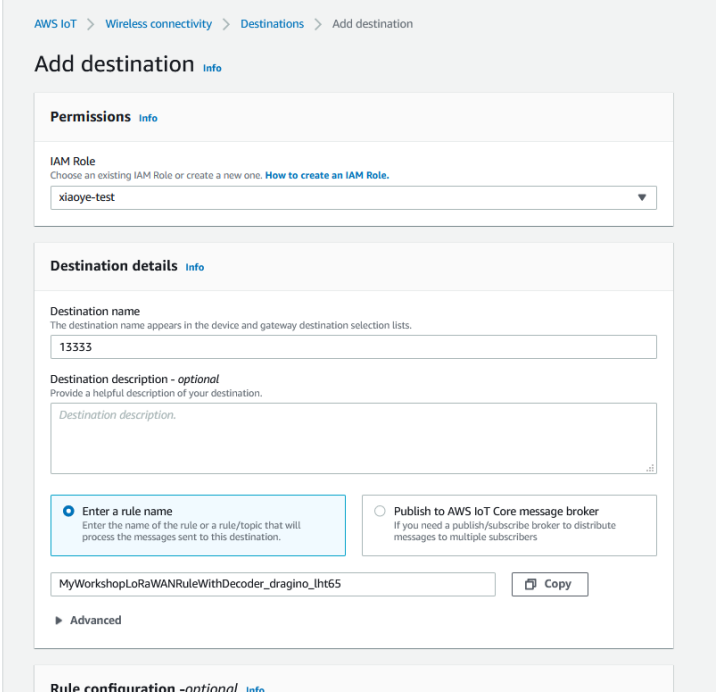
Step 1: Find the IoT Rule MyWorkshopLoRaWANRuleWithDecoder

Please put the IoT Rule name prefix **MyWorkshopLoRaWANRuleWithDecoder_** into the search field and click on the search symbol. The rule named **MyWorkshopLoRaWANRuleWithDecoder_<Decoder name>** should appear:

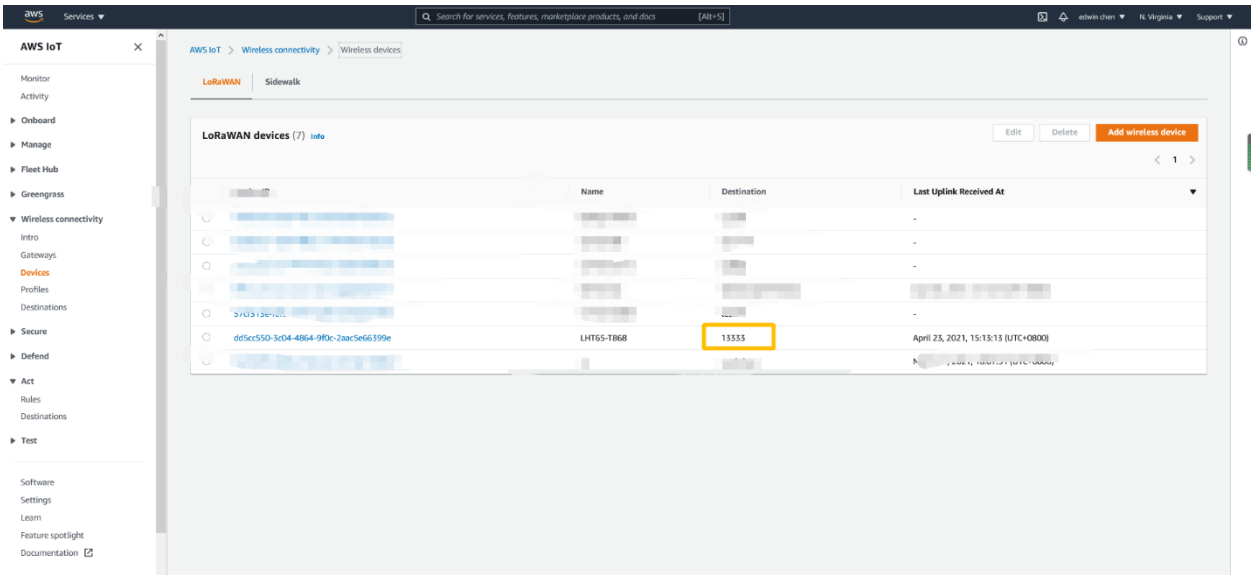


Step 2: Create a Destination with IoT Rule

(MyWorkshopLoRaWANRuleWithDecoder_dragino_lht65)



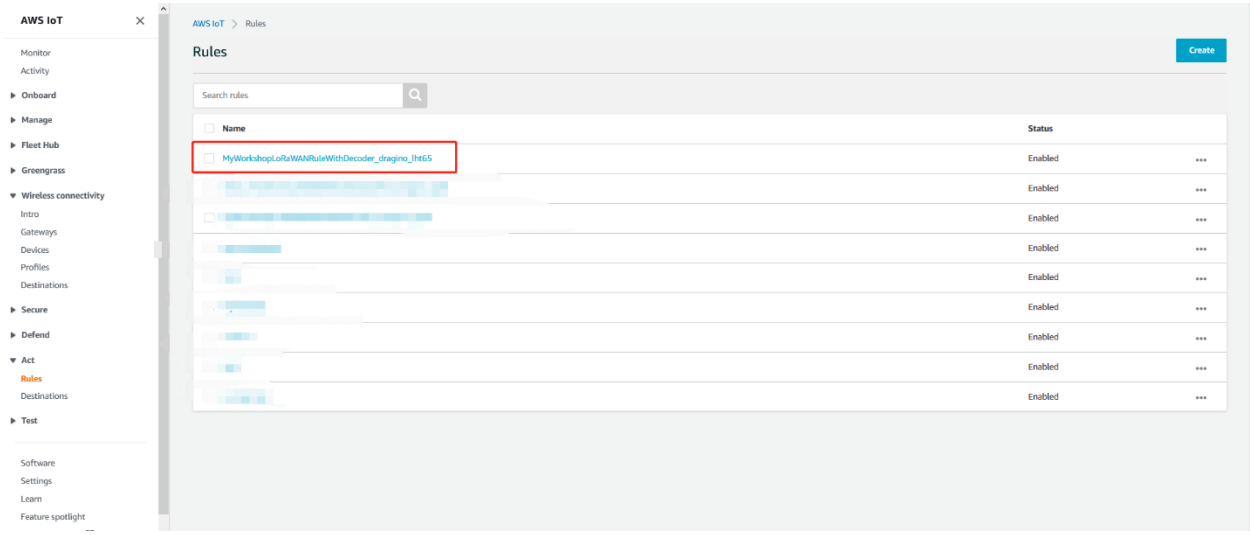
Step 3: Update the destination to the device



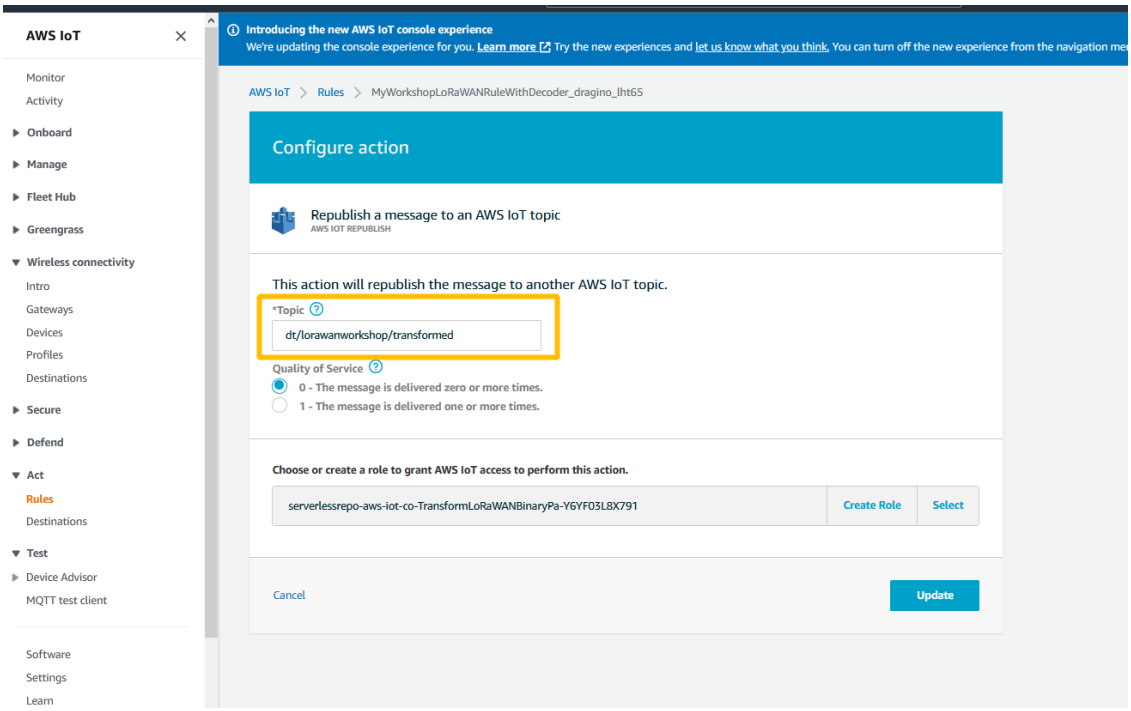
Step 4 Check the payload

Approach A with MQTT

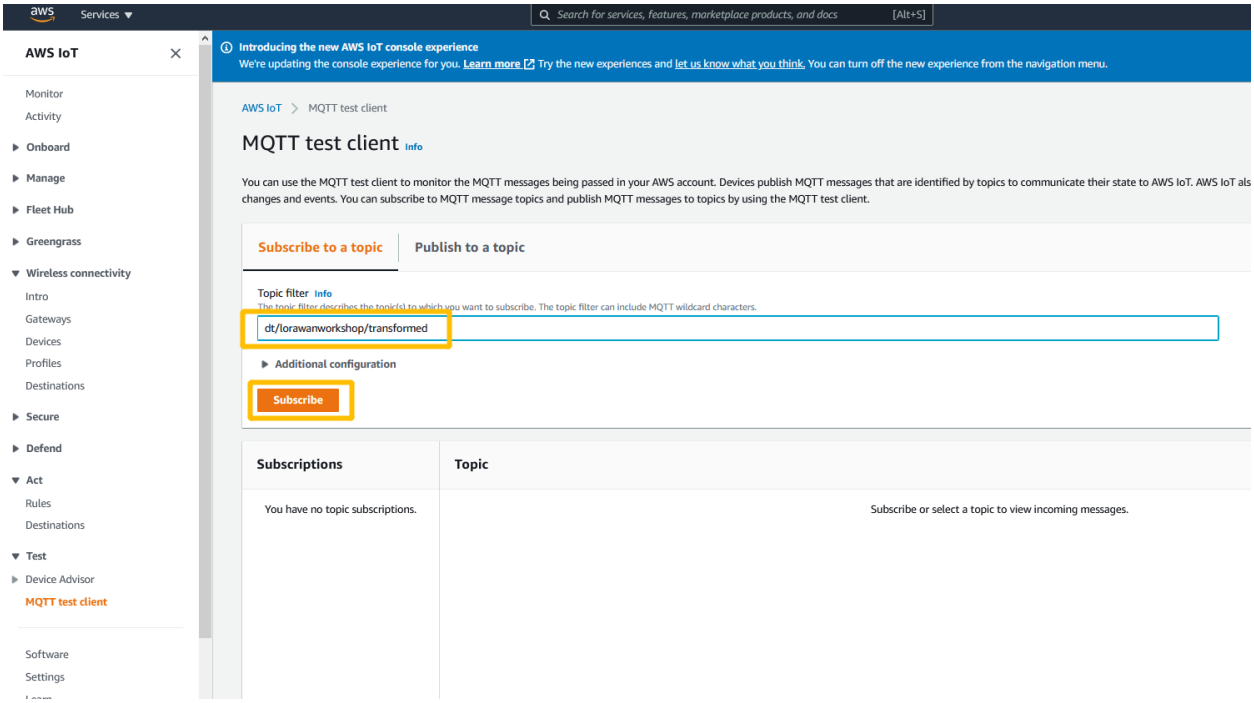
click the rule



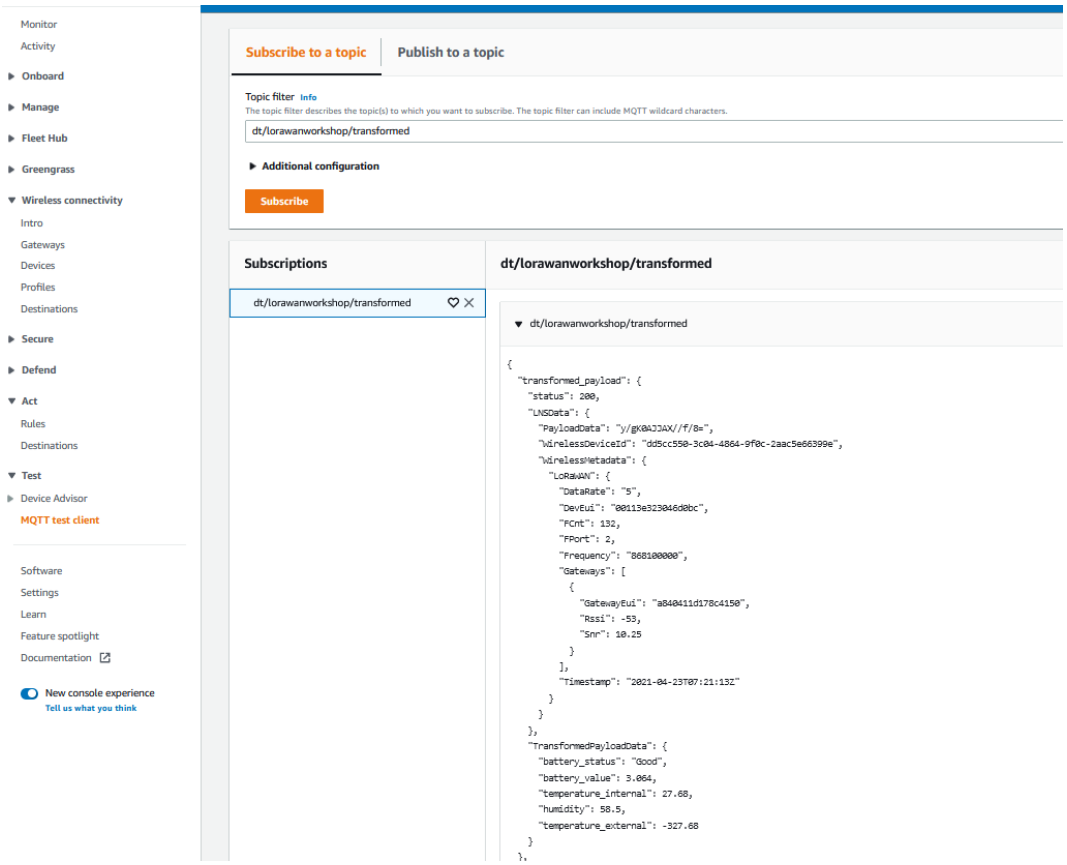
Copy the Topic name



Open MQTT test client and subscribe Topic

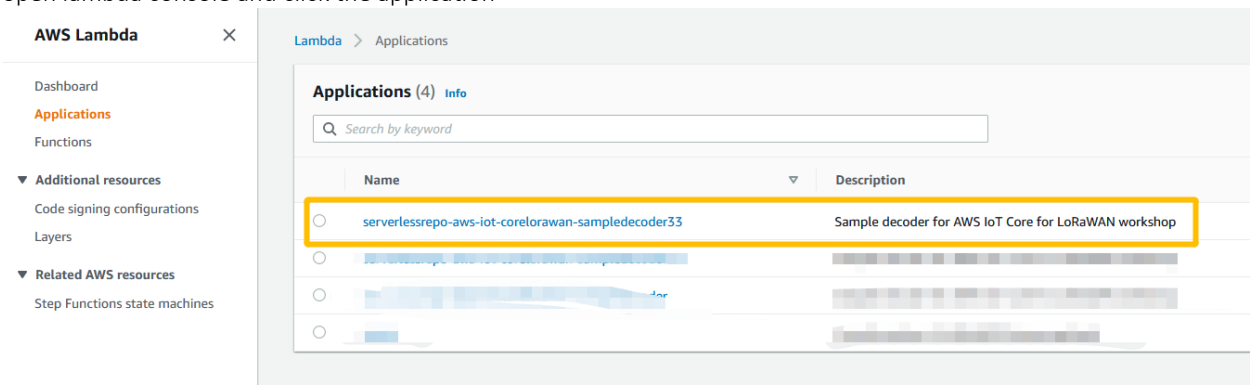


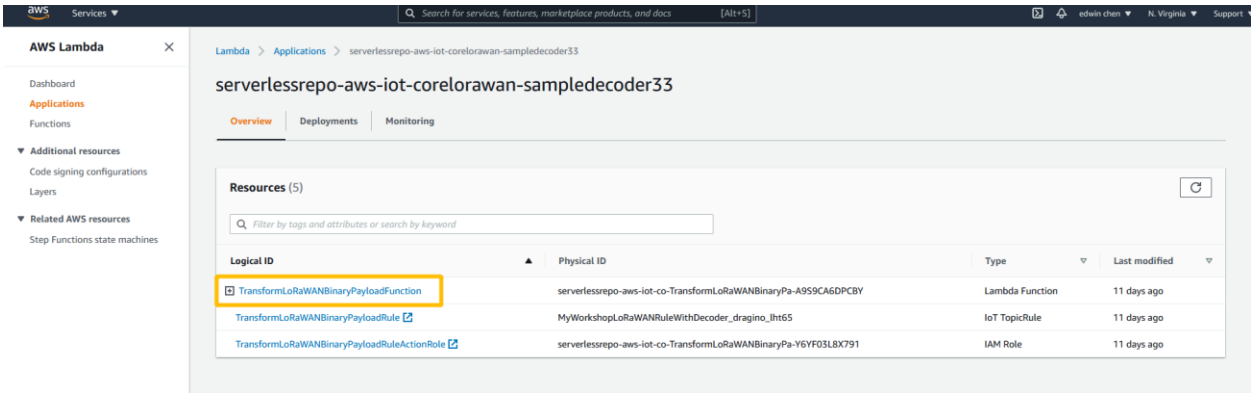
Get the payload



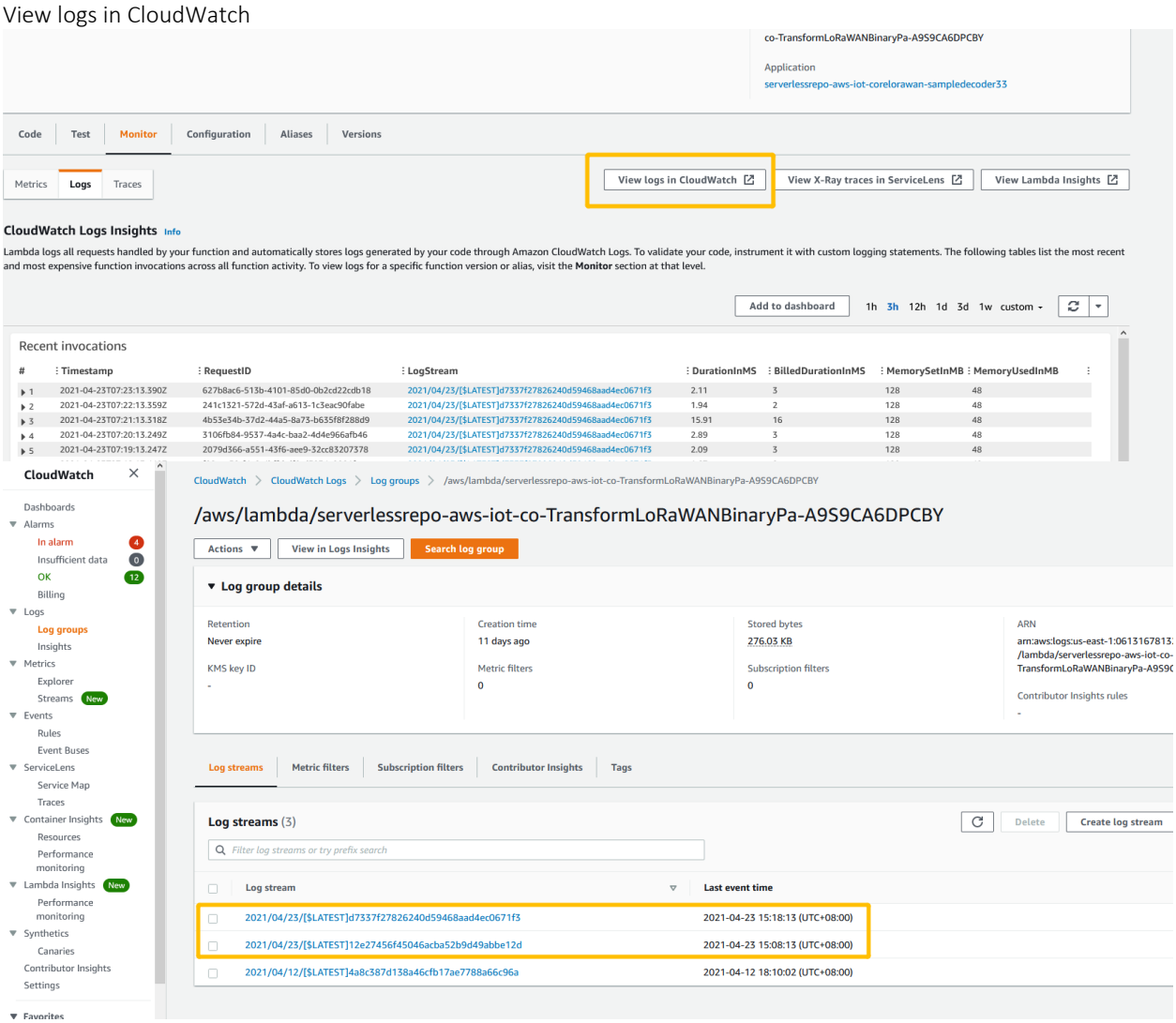
Approach B with Lambda

open lambda console and click the application





View logs in CloudWatch



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:

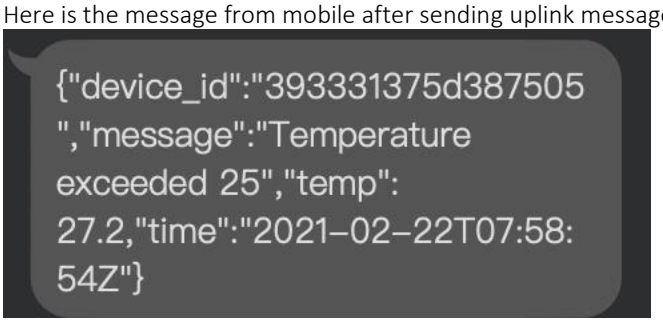
```
SELECT DevEUI as device_id, "Temperature exceeded 25" as message,
Alert_Temp as temp, Humidity as humidity, Timestamp as time FROM
'project/sensor/decoded' where Alert_Temp > 25
```

- 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.

7.3.2 Test the rule for Amazon SNS notification

After adding the rule for Amazon SNS notification, we should receive text message when hitting the event.

- Send message from end Device using AT command: **at+send:lora:1:01670110**
- Here is the message from mobile after sending uplink message.



7.4 Send Downlink Payload

This section shows how to send downlink payload from AWS IoT LoRaWAN Server to end Device.

Please follow the instructions on How to Send Downlink Payload.

7.5 IoT Analytics

7.5.1 Introduction

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

7.5.2 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.

7.5.3 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**

7.5.4 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 IoTAnalyticsRule
- Choose **Create data source**, and then choose **Visualize**
- Select dataset created, then select **Refresh** or **Schedule Refresh** for periodic refresh of dataset.

7.6 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:

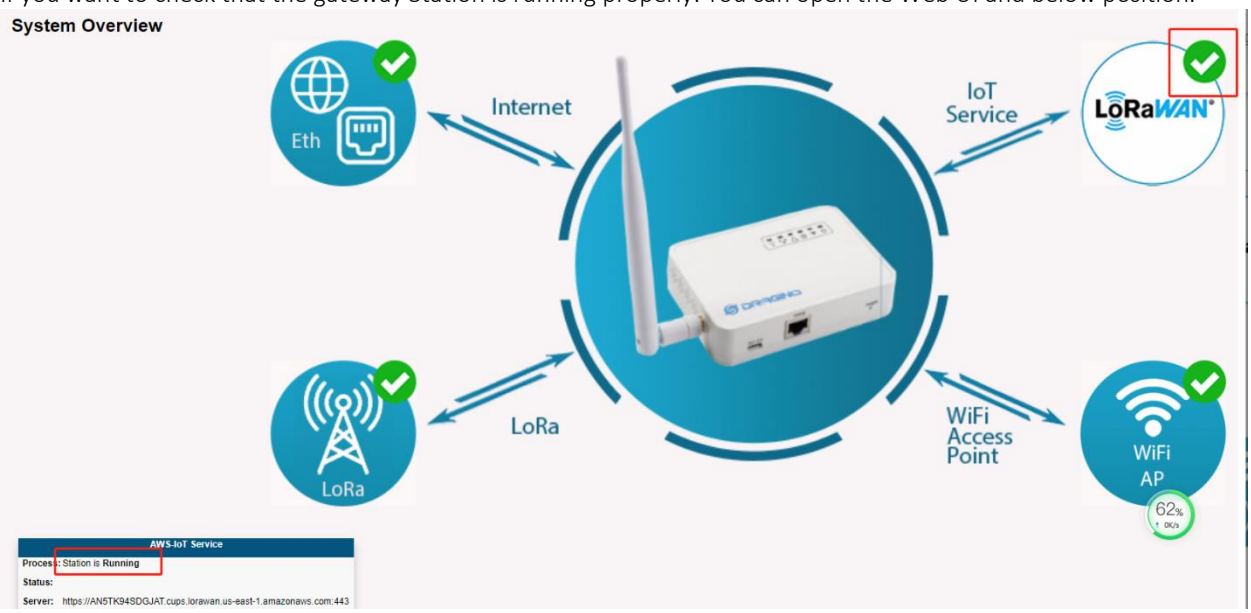
- Go to the AWS IoT Analytics console
- Choose **Data sets**

- 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

8.1 How to check the gateway is running properly to connect AWS-IoT

If you want to check that the gateway Station is running properly. You can open the Web UI and below position.



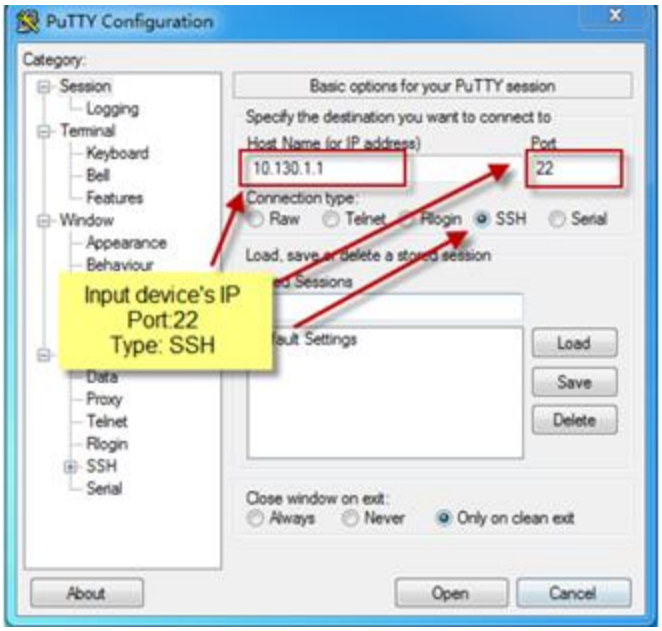
8.2 How to get Station Log

User can access to the Linux console via SSH protocol. Make sure your PC and the LIG16 is in the same network, then use a SSH tool (such as putty, SecureCRT) to access it. Below are screenshots:

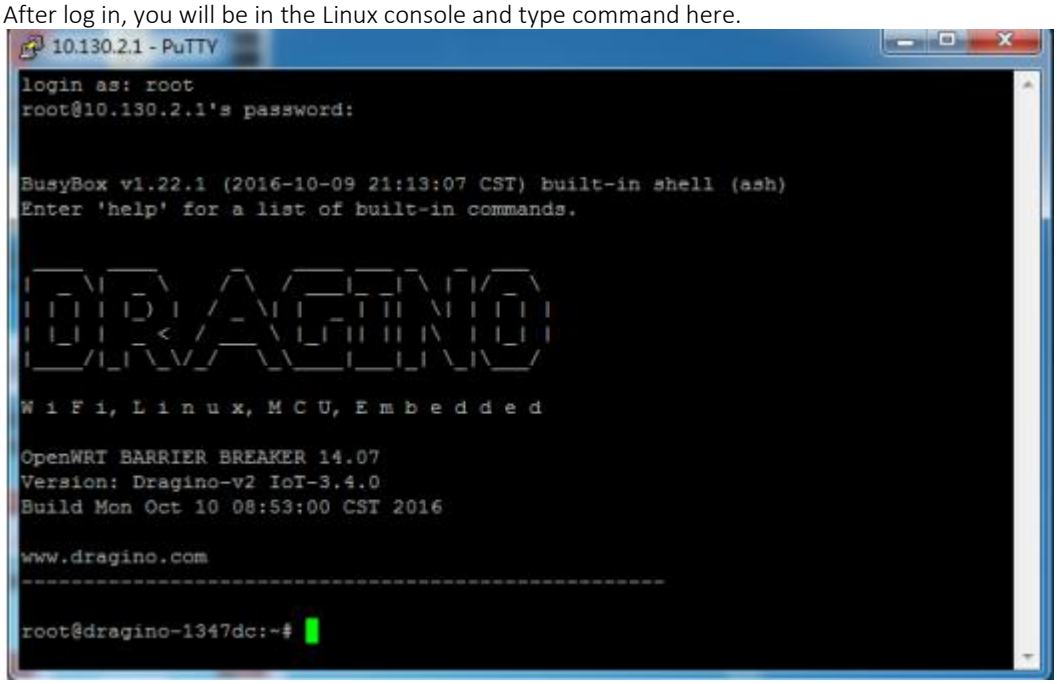
If you want to check the Station Log, please run this command : cat /var/iot/station.log

8.3 Access the gateway Linux console

IP address: IP address of LIG16
Port: 22 or 2222
User Name: root
Password: dragino (default)



After log in, you will be in the Linux console and type command here.



If you want to check the Station Log, please run this command: cat /var/iot/station.log

```
root@dragino-1ec39c:~# cat /var/iot/station.log
2021-03-22 06:12:49.305 [SYS:INFO] Logging     : /var/iot/station.log (maxsize=10000000, rotate=3)
2021-03-22 06:12:49.306 [SYS:INFO] Station Ver : 2.0.6(mips-openwrt/dragino) 2021-03-16 04:13:21
2021-03-22 06:12:49.306 [SYS:INFO] Package Ver : (null)
2021-03-22 06:12:49.306 [SYS:INFO] proto EUI   : a840:41ff:ff1e:c39c    (station.conf)
2021-03-22 06:12:49.306 [SYS:INFO] prefix EUI  : ::1    (builtin)
2021-03-22 06:12:49.306 [SYS:INFO] Station EUI : a840:41ff:ff1e:c39c
2021-03-22 06:12:49.307 [SYS:INFO] Station home: ./     (builtin)
2021-03-22 06:12:49.307 [SYS:INFO] Station temp: /var/tmp/      (builtin)
2021-03-22 06:12:49.313 [SYS:INFO] DAEMON: Station process 12723 started...
2021-03-22 06:12:49.525 [TCE:INFO] Starting TC engine
2021-03-22 06:12:49.526 [TCE:ERRO] No TC URI configured
2021-03-22 06:12:49.527 [CUP:INFO] Starting a CUPS session in 0 seconds.
2021-03-22 06:12:49.527 [TCE:INFO] Router rejected or retry limit reached. Invoking CUPS.
2021-03-22 06:12:49.527 [TCE:INFO] Terminating TC engine
2021-03-22 06:12:49.527 [CUP:INFO] Starting a CUPS session now.
2021-03-22 06:12:49.528 [CUP:INFO] Connecting to CUPS ... https://AN5TK94SDGJAT.cups.lorawan.us-east-1.amazonaws.com:443 (try #1)
2021-03-22 06:12:49.531 [any:INFO] ./cups.trust:
cert. version     : 3
serial number     : 06:7F:94:57:85:87:E8:AC:77:DE:B2:53:32:5B:BC:99:8B:56:0D
issuer name       : C=US, O=Amazon, CN=Amazon Root CA 1
subject name      : C=US, O=Amazon, OU=Server CA 1B, CN=Amazon
issued  on        : 2015-10-22 00:00:00
expires on        : 2025-10-19 00:00:00
signed using      : RSA with SHA-256
RSA key size      : 2048 bits
basic constraints : CA=true, max_pathlen=0
key usage         : Digital Signature, Key Cert Sign,2021-03-22 06:12:49.589 [any:INFO] ./cups.crt:
cert. version     : 3
serial number     : 49:83:CA:64:97:3C:19:27:40:26:6E:EE:65:B7:30:A7:87:71:BA:F8
issuer name       : OU=Amazon Web Services O=Amazon.com Inc. L=Seattle ST=Washington C=US
subject name      : CN=AWS IoT Certificate
issued  on        : 2021-03-22 06:09:43
expires on        : 2049-12-31 23:59:59
signed using      : RSA with SHA-256
RSA key size      : 2048 bits
basic constraints : CA=false
key usage         : Digital Signature
2021-03-22 06:12:49.589 [AIO:INFO]
2021-03-22 06:12:53.625 [CUP:VERB] Retrieving update-info from CUPS https://AN5TK94SDGJAT.cups.lorawan.us-east-1.amazonaws.com:443...
```

If you are monitoring the Station Log in real time, first please run this command : cd /etc/station; station -f

```
BusyBox v1.28.3 () built-in shell (ash)

 _____ _____ _____ _____ _____
|  D  |  R  |  A  |  G  |  I  |  N  |  O  |
|_____|_____|_____|_____|_____|_____|_____|
w i F i, L i n u x, M C U, E m b e d d e d

OpenWRT 18.06
Version: Dragino-v2 lgw-5.4.1615882321
Build Tue Mar 16 16:12:01 CST 2021

www.dragino.com
-----------------------------------------------------

root@dragino-1ec39c:~#  cd /etc/station/; station -f
Killing process 12045
2021-03-22 09:17:34.950 [SYS:INFO] Logging     : stderr (maxsize=10000000, rotate=3)
2021-03-22 09:17:34.950 [SYS:INFO] Station Ver : 2.0.6(mips-openwrt/dragino) 2021-03-16 04:13:21
2021-03-22 09:17:34.950 [SYS:INFO] Package Ver : (null)
2021-03-22 09:17:34.950 [SYS:INFO] proto EUI   : a840:41ff:ff1e:c39c    (station.conf)
2021-03-22 09:17:34.950 [SYS:INFO] prefix EUI  : ::1    (builtin)
2021-03-22 09:17:34.950 [SYS:INFO] Station EUI : a840:41ff:ff1e:c39c
2021-03-22 09:17:34.950 [SYS:INFO] Station home: ./     (builtin)
2021-03-22 09:17:34.950 [SYS:INFO] Station temp: /var/tmp/      (builtin)
2021-03-22 09:17:35.154 [TCE:INFO] Starting TC engine
2021-03-22 09:17:35.165 [any:INFO] ./tc.trust:
cert. version     : 3
serial number     : 06:7F:94:57:85:87:E8:AC:77:DE:B2:53:32:5B:BC:99:8B:56:0D
issuer name       : C=US, O=Amazon, CN=Amazon Root CA 1
subject name      : C=US, O=Amazon, OU=Server CA 1B, CN=Amazon
issued  on        : 2015-10-22 00:00:00
expires on        : 2025-10-19 00:00:00
signed using      : RSA with SHA-256
RSA key size      : 2048 bits
basic constraints : CA=true, max_pathlen=0
key usage         : Digital Signature, Key Cert Sign, C2021-03-22 09:17:35.252 [any:INFO] ./tc.crt:
cert. version     : 3
serial number     : 80:A8:6B:16:50:91:8E:A3:62:E0:DD:4A:F3:9C:04:3A:07:54:A0:76
issuer name       : OU=Amazon Web Services O=Amazon.com Inc. L=Seattle ST=Washington C=US
subject name      : CN=AWS IoT Certificate
issued  on        : 2021-03-22 06:10:58
expires on        : 2049-12-31 23:59:59
signed using      : RSA with SHA-256
RSA key size      : 2048 bits
basic constraints : CA=false
key usage         : Digital Signature
2021-03-22 09:17:35.252 [AIO:INFO]
2021-03-22 09:17:35.505 [TCE:INFO] Connecting to INFOS: wss://AN5TK94SDGJAT.gateway.lorawan.us-east-1.amazonaws.com:443
2021-03-22 09:17:35.506 [CUP:INFO] Starting a CUPS session in 0 seconds.
2021-03-22 09:17:35.507 [CUP:INFO] Starting a CUPS session now.
2021-03-22 09:17:35.508 [CUP:INFO] Connecting to CUPS ... https://AN5TK94SDGJAT.cups.lorawan.us-east-1.amazonaws.com:443 (try #1)
2021-03-22 09:17:35.509 [any:INFO] ./cups.trust:
cert. version     : 3
serial number     : 06:7F:94:57:85:87:E8:AC:77:DE:B2:53:32:5B:BC:99:8B:56:0D
issuer name       : C=US, O=Amazon, CN=Amazon Root CA 1
subject name      : C=US, O=Amazon, OU=Server CA 1B, CN=Amazon
issued  on        : 2015-10-22 00:00:00
```

Note that if you log out then station will log out as well, requiring you to type the command cd /etc/station/; station -d.

9 Troubleshooting

9.1 For resolving common or potential problems

User gateway may not start Station properly and therefore cannot connect to AWS
Please check: Is GWID consistent with AWS-Gateway EUI?
Re-upload the certificate and Save&Apply it again

9.2 Firmware version

Firmware version must be lgw--build-v5.4.1615882321-20210316-1613 or newer.
See here for how to check version.

9.3 Contact Dragino for Directly Support

If the above debugging is not possible

Please send mail to : support@dragino.com

10 OTA Updates

Currently not supported.