AWS IoT Core for LoRaWAN

Sentrius™ RG1xx and RS1xx

Version 2.0

Revision History

Version	Date	Notes	Contributor	Approver
1.0	5 Jan 2021	Initial Release	Greg Leach	Chris Boorman
1.1	12 Jan 2021	Add definition of supported hardware	Chris Boorman	Jonathan Kaye
1.2	9 Mar 2021	Added further entries to Glossary section. Corrected filename references in sections 9.1.1 and 9.2.1. Clarified section 6.4.3 for source of Destination Role. Added further detail to section 6.4.3 to describe creation of the root MQTT topic for other Actions and Services. Added further detail to Section 9 to clarify the initial source of data for created Rules. Added Section 9.1.3 to describe update of the Role used to grant Lambda code access to the IoT Core. Updated Figure 39 and Figure 75 to emphasise Lambda Engine boundary and the requirement for the access Policy to be updated via IAM.	Greg Leach	Chris Boorman
1.3	29 July 2023	Updated for changes made to AWS IoT Core for LoRaWAN user interface. Section 6.3 updated to reflect changes to IoT Core for LoRaWAN interface. Section 6.4 updated to reflect changes to IoT Core for LoRaWAN interface. Section 7.3 updated to clarify usage of files downloaded from AWS for storage in gateway. Section 9 updated to reflect changes to IoT Core for LoRaWAN interface. Added section 14 to detail definition of Device profiles for RS1xx devices.	Greg Leach	Senthooran Ragavan
2.0	2 May 2025	Ezurio rebranding	Sue White	Dave Drogowski

Contents

1 Introduction

This application note describes the steps involved in integrating Ezurio's RG1xx series of gateways and RS1xx series of sensors with AWS IoT Core for LoRaWAN. It is intended to be referred to in conjunction with the RG1xx User Guide [A] & RG1xx User Guide (LTE) [B], which further describe RG1XX related activities, the RS1XX Configuration Guide [O], RS1xx Open/Closed & Internal Temperature/Humidity Sensor User Guide [P] & RS1xx External Temperature Sensor User Guide [Q], which further describe RS1xx related activities, and the AWS IoT Core for LoRaWAN User Guide [C], which describes usage of AWS IoT Core for LoRaWAN.

IMPORTANT! A minimum firmware version of 93.8.5.25 is required for the RG1xx gateway. This is to ensure the version of Semtech Basics Station meets the minimum required (v2.0.5) by AWS IoT Core for LoRaWAN.
See Section 7.2, "Set Up Software" for guidance.

It should also be noted, at this time the **RG191+LTE** gateway (part numbers: 450-00107-K1 / 450-00109-K1) **does not support AWS IoT Core for LoRaWAN**. Support will be included as part of a new GA update for those gateway models set for release Q1 2021.

1.1 Naming Conventions

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

1.2 Glossary

Term	Definition
ARN	Amazon Resource Number
AWS	Amazon Web Services
CUPS	Configuration and Update Service
DHCP	Dynamic Host Configuration Protocol
EIRP	Effective Isotropic Radiated Power
DNS	Domain Name Server
EUI	Extended Unique Identifier
GA	General Availability
IAM	Identity and Access Management
IoT	Internet of Things
LAN	Local Area Network
LNS	LoRaWAN Network Server
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
LRC	Long Range Controller
LTE	Long Term Evolution, 4G/5G based cellular communications specification
MQTT	Message Queuing Telemetry Transport
OTA	Over the Air
SLAAC	State Less Address Auto Configuration
URL	Universal Resource Locator

2 Gateway Overview

The Sentrius RG1xx LoRaWAN-Enabled Gateway (Figure 1) is the ultimate in secure, scalable, robust LoRaWAN solutions. Data can be gathered from as far as 10 miles via LoRaWAN, then synchronized to the cloud via Wi-Fi / Ethernet, or LTE in the US with the LTE version. The RG1xx gives full ownership over a network, adding multi-protocol connectivity to sensors and devices to create actionable IoT intelligence.

Figure 1: Sentrius RG1XX Gateway

3 Gateway Hardware Description

3.1 Datasheet

Refer to [A] & [B] and the RG1xx Product Brief [E] and RG1XX LTE Product Brief [F].

3.2 Standard Kit Contents

Each RG1xx ships with 1 x region specific LoRa antenna (868/915/923MHz), 2 x 2.4/5 GHz antenna for Wi-Fi connectivity, an external DC power supply and an Ethernet cable.

3.3 User Provided Items

An AWS account is required for connectivity to AWS IoT Core for LoRaWAN.

3.4 Third Party purchasable items

Endpoint devices are required as data sources for the gateway. Ezurio recommends our Sentrius RS1xx range of sensors. Refer to the Product Briefs for the External RTD Temperature Probe [G], the External Temperature Sensor [H], the Integrated Temperature & Humidity Sensor [I] and the Open/Closed Sensor with Integrated Temperature & Humidity Sensor [J] for further details.

3.5 Additional Hardware References

A complete list of available certifications for the RG1xx Gateway is available from the RG1xx product page [K] under "Documentation."

4 Sensor Overview

The Sentrius™ RS1xx sensors (Figure 2) enable battery-powered, local and wide-area sensor applications using LoRaWAN and Bluetooth 4.2. The RS1xx are small, rugged, and easily configurable sensors making it easier than ever to monitor environmental data with a network of sensors.

All variants are powerful yet power-conscious sensors, configurable to transmit infrequently and last for years on the same set of two replaceable AA batteries. A sensor network can be easily created that provides miles of coverage, easily configurable via a smartphone or tablet.

Figure 2: Sentrius RS1xx Sensor

5 Sensor Hardware Description

5.1 Datasheet

Refer to reference [G], [H], [I] and [J].

5.2 Standard Kit Contents

Each sensor is shipped with a pair of Alkaline AA batteries. Externally connected sensors must be ordered separately.

5.3 User Provided Items

The user is responsible for mounting the sensor. A BLE 4.2 capable mobile device is required for local device configuration.

5.4 Third Party purchasable items

LoRa gateway devices are required as gathering points for sensor data. Laird recommends its RG1xx range of gateways, refer to section **Error! Reference source not found.**, Gateway Overview.

5.5 Additional Hardware References

A complete list of available certifications for the RS1xx range of sensors is available from the RS1xx product page [R] under '"Documentation."

6 Setup your AWS account and Permissions

If you don't have an AWS account, refer to the instructions in the AWS setup guide [L]. The relevant sections are "Sign up for an AWS account" and "Create a user and grant permissions."

6.1 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 [X].

6.2 Set up Roles and Policies in IAM

6.2.1 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
- Click **Create role**.
- On the Create Role page, choose *Another AWS account*.
- For Account ID, enter your account ID.
- Click **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
        - Click **Policies** on the navigation pane.
        - Click **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": "*"
        }
```

```
    ]
}
```

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

- Click **Next: Tags**, and then click **Next: Review**.
- In Role name, enter IoTWirelessGatewayCertManagerRole, and then click **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 click **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": {}
    }
  ]
}
```

Click **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 [M] and Security Best practices [N].

6.2.2 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
- Click **Policies** in the navigation pane.
- Click **Create Policy**. Then click 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": "*"
      }
    ]
}
```

- Click **Next: Tags.**
- Click **Next: Review** to open the Review page.
- 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, click **Roles** from the navigation pane to open the **Roles** page.
- Click **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.
- Click **Next: Tags**.
- Click **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.
- Click **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 the Summary page.
- Click the **Trust relationships** tab to navigate to the Trust relationships page.
- Click **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*.

6.3 Add the Gateway to AWS IoT

Note: The account region must be set to us-east-1 or us-west-2 for the **LPWAN devices** menu item to be displayed.

6.3.1 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 (DevEUI) 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.
- Note that Semtech Basics Station v2.0.5 and greater is required.

- Note that RG1XX firmware v93.8.5.25 and greater is required.

6.3.2 Add the LoRaWAN Gateway

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

- Go to the AWS IoT Core console.
- Under **LPWAN devices**, click **Gateways**.
- Click **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 do not recommend you leave it blank.
- Click **Add gateway**.
- On the Configure your Gateway page, find the section titled Gateway certificate.
- Click **Create certificate**.
- Once the Certificate created and associated with your gateway message is shown, click **Download certificates** to download the certificate (xxxxx.cert.pem) and private key (xxxxxx.private.key).
- In the section Provisioning credentials, click **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.
- Click **Submit** to add the gateway.

6.4 Add a LoRaWAN Device to AWS IoT

Note: Refer to section 14 for RS1xx specific setup information.

6.4.1 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:
  - AU915
  - EU868
  - EU433
  - CN470
  - CN779
  - RU864
  - KR920
  - IN865
  - US915
  - AS923-1
  - AS923-2
  - AS923-3
  - AS923-4
- MAC Version. This must be one of the following:
  - V1.0.2
  - v1.0.3
  - v1.1
- Regional parameters version. The following Regional parameters versions are supported:
  - LoRaWAN v1.0.1
  - Regional Parameters v1.0.2rB
  - Regional Parameters v1.0.3rA
  - Regional Parameters v1.1rA
  - RP002-1.0.0
  - RP002-1.0.1
- The device join mechanism. The following are supported:
  - OTAA v1.0x and OTAA v1.1.
  - ABP v1.0x and ABP v1.1.

- The MaxEIRP for the device.

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, NwkSkey, FNwkSIntKey, SNwkSIntKey, AppSKey

6.4.2 Verify Device & Service 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 Core console. In the navigation pane, click **LPWAN devices**.
- 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 a Ezurio US915 RS1xx as an example, the values are:
  - MacVersion 1.0.2
  - 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.
  - UlRate60
  - UlBucketSize4096
  - DlRate60
  - DlBucketSize4096
  - AddGwMetadatatrue
  - DevStatusReqFreq24
  - DrMax15
  - TargetPer5
  - MinGwDiversity1

Proceed only if you have a Device and Service profile that will work for you.

6.4.3 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 [X].

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

- Navigate to the AWS IoT Core console. In the navigation pane, click **LPWAN devices**, and then **Destinations**.
- Click **Add Destination**.
- On the Add destination page, in the **Permissions** section, select **Select an existing service role**, then select the IAM Role created in section 6.2.2 from the drop-down.
- **Under Destination** details enter a suitable name as the **Destination** name, and an appropriate description under Destination description – optional. It should be considered the Destination will be entry point into AWS for a group of devices, with naming needing to reflect this.
- The **Rule Name** and **Rule** configuration sections are used to configure the Rule invoked AWS IoT Core side when data is received from sensors. The name chosen should reflect this.
- Upon having entered the **Rule Name**, click **Copy**.
- Click **Create Rule**. This allows definition of the Rule used as the entry point for incoming sensor data.
- In the **Name** field, enter the Name copied from the previous step.

- Click **Next**, then set the Rule Query Statement as follows:

```
SELECT * FROM 'iot/topic'
```

- Click **Next**, then select **Republish to AWS IoT Topic** and enter a suitable name for the *Topic* (e.g. 'SensorOutput'). Note the Topic name for use later – this is the root Topic that is used to pass sensor data to other AWS Rules and Services.
- Create a **Role** for the Action.
- Click **Add Action**.
- Click **Create Rule** to finalise creation.
- Click **Add Destination**. You will see a message confirming "Destination added", indicating the destination has been successfully added.
- This Destination can be used by multiple sensors. Messages from all sensors using this Destination will be routed to the MQTT Topic created.

Refer to Figure 3 for an example of how the Destination and associated Rule defines how sensor data is routed to a root MQTT topic.

The user network consists of US sensors and gateways, and EU sensors and gateways. Messages from the US Sensors are routed to one Destination, and those from the EU Sensors to another by the AWS IoT Core Rules Engine. This invokes the appropriate Rule (EU Sensor Routing Rule for EU sensors, and US Sensor Routing Rule for US sensors).

Two MQTT topics, Root EU Sensor Topic for EU sensors and Root US Sensor Topic for US sensors, are then published to. This results in separate data sets for the two sensor types.

Note that data published to the Root Topic is unprocessed. It contains the raw payload data, in addition to gateway information. For meaningful data to be made available, the Root Topic must be used as the source for further Actions and Services. This is described further in Section 9.

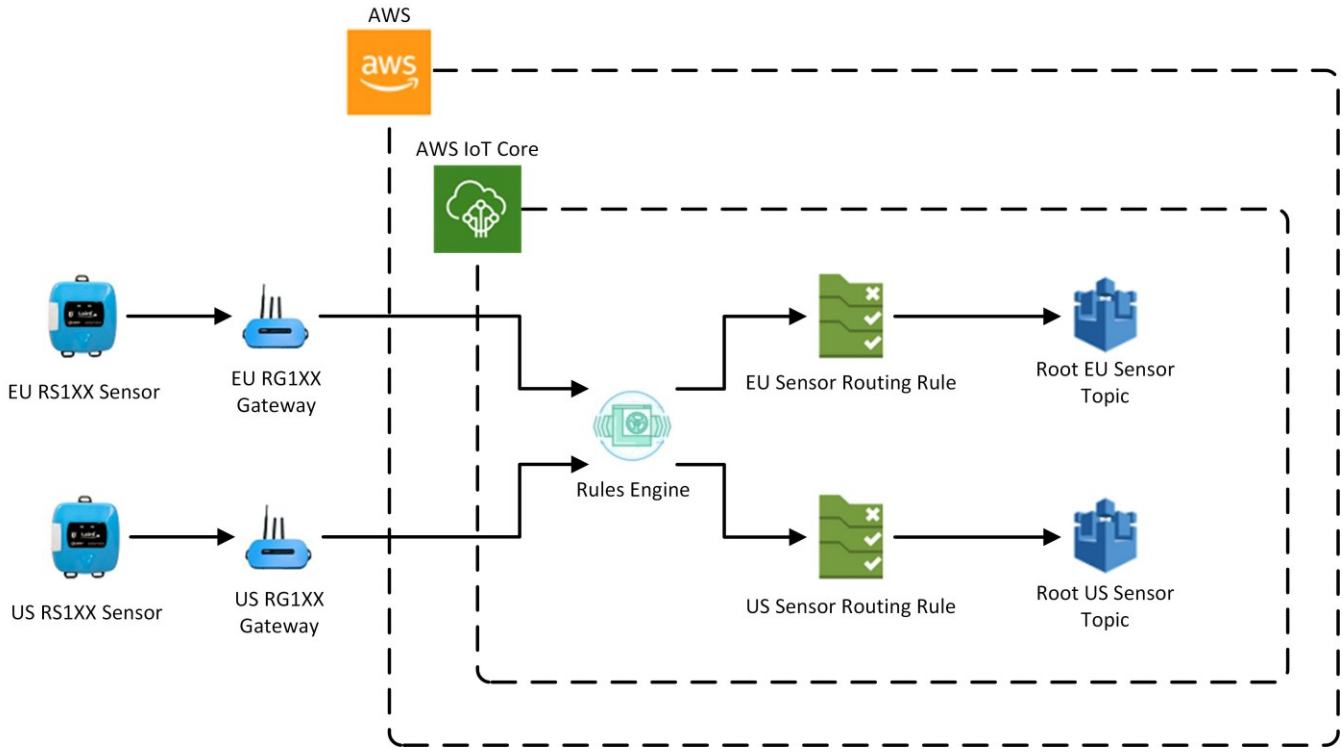

Figure 3: Root MQTT Topic via Destination Rule

6.4.4 Register the Device

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

- Go to the AWS IoT Core console.
- Click **LPWAN devices** in the navigation panel on the left.
- Click **Devices**.
- Click **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 section 6.4.2 Verify Device & Service Profiles).

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

7 Set Up the Gateway

7.1 Set up hardware

The following describes the steps required to setup the RG1xx Gateway. Figure 4 shows the hardware features of the gateway.

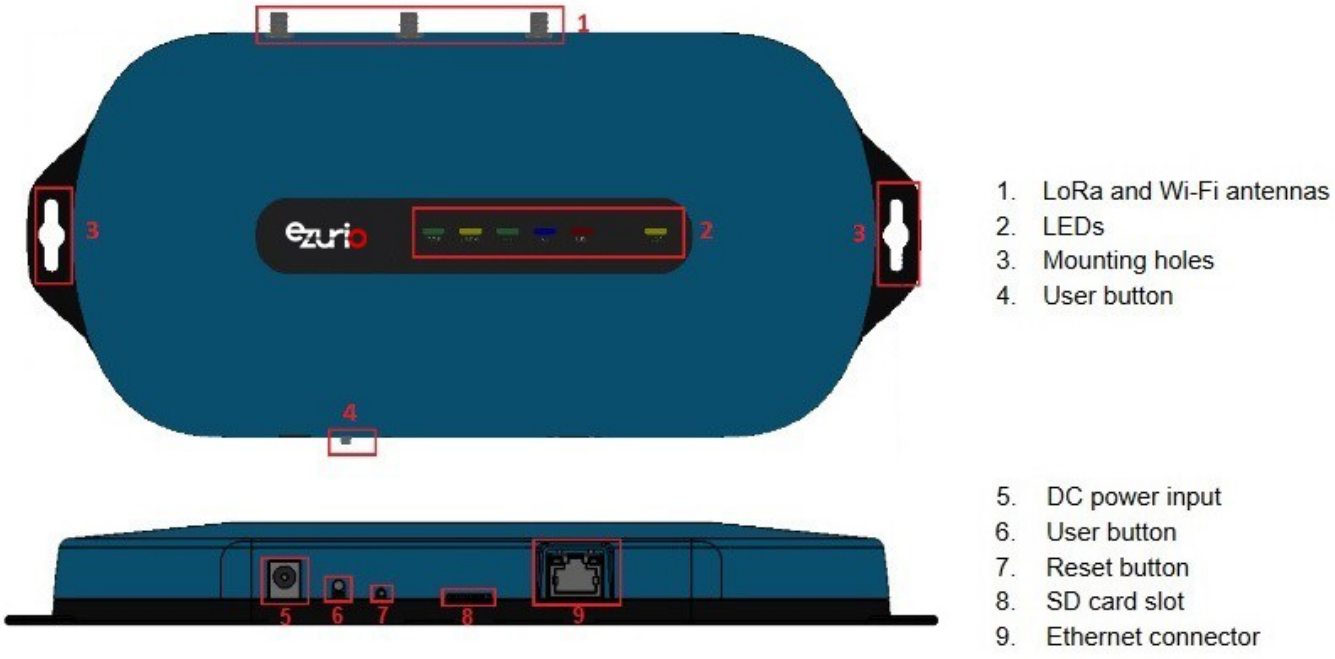

Figure 4: Sentrius RG1xx gateway hardware features

7.1.1 Physical Connectivity

The supplied antennae are first connected to the gateway before power-up. Figure 5 indicates the location of the LoRa and Wi-Fi antennae.

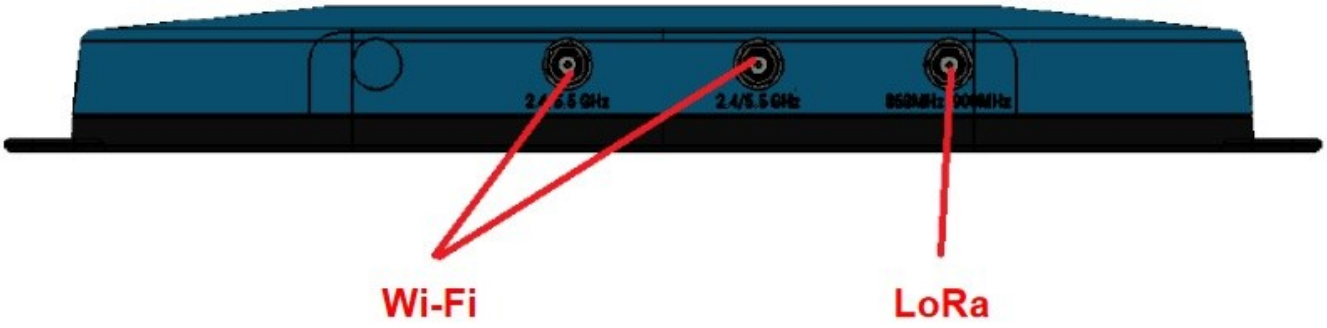

Figure 5: RG1xx antennae connectivity

As shown in Figure 6, the external DC power supply must be connected (1) and mains power provided. For Ethernet connectivity, the supplied Ethernet cable is connected (2) and to the end user router (3).

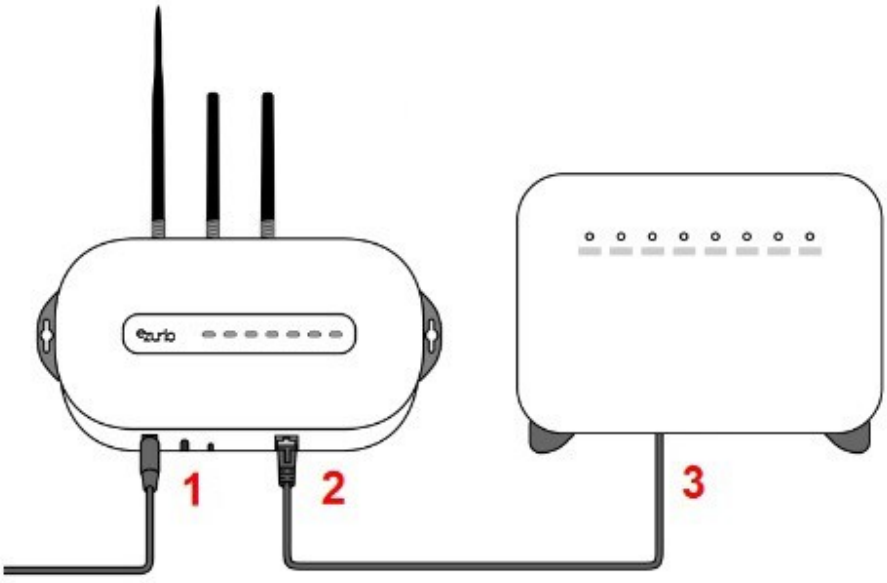

Figure 6: Gateway physical connectivity

7.1.2 Gateway LEDs

The LED array visible on the front panel of the RG1XX gateway is shown in Figure 7. Table 1 describes the purpose of each LED.

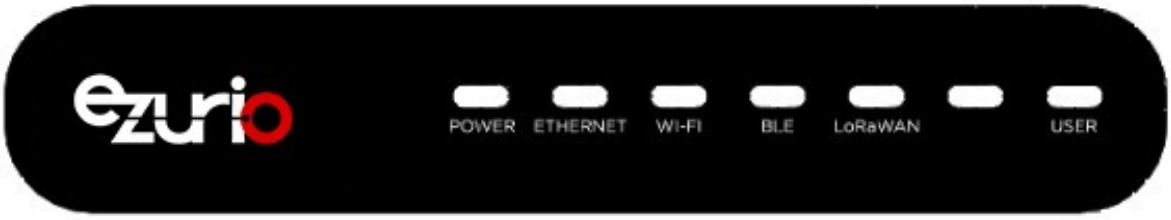

Figure 7: RG1XX LEDs

Table 1: RG1XX LEDs

Label	Purpose
Power	Illuminated when power is applied.
Ethernet	Off when Ethernet hardware is disabled.

Label	Purpose
	Illuminated when Ethernet hardware is initialized. Flashes when Ethernet communications are in progress.
Wi-Fi	Off when WiFi hardware is disabled. Illuminated when WiFi hardware is initialized. Flashes when WiFi communications are in progress.
BLE	Illuminated when BLE hardware is initialized. Flashes when BLE communications are in progress.
LoRaWAN	Illuminated when LoRa hardware is initialized. Flashes when LoRa communications is in progress.
User	Reserved for future use.

7.1.3 Logging into the gateway

To log into the gateway web interface, complete the following steps.

Determine the last three bytes of the gateway's Ethernet MAC address, found on the label on the bottom of the gateway as shown in Figure 8 with the last three bytes highlighted.

Figure 8: Determining the gateway Ethernet MAC address

Each gateway exposes an HTTP web server, with a DNS being used to create a unique address for each gateway. This takes the form https://RG1xxXXXXXX.local, where XXXXXX are the last three bytes of the gateway MAC address. For example, for a gateway with 29378B as the last three bytes of its MAC address, the address for the gateway would be https://RG1xx29378B.local.

Enter the gateway address into a web browser and confirm. A dialog of the form shown in Figure 9 is first shown. Click **Yes** to proceed.

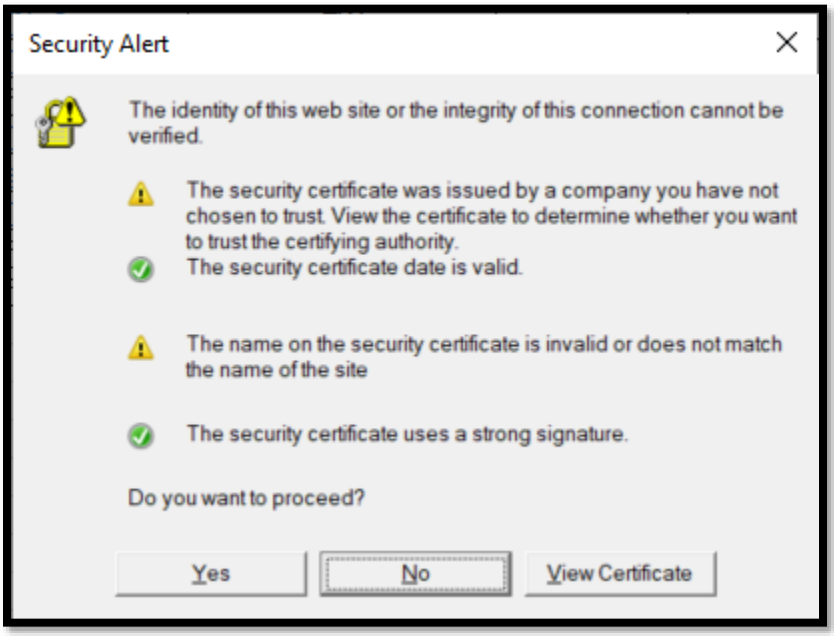

Figure 9: Dialog shown when first opening gateway web interface

The gateway web interface log in page appears as shown in Figure 10. Enter your credentials if you've changed the default username and password. The default credentials are as follows:

Username: sentrius

Password: RG1xx

Then click **Login**.

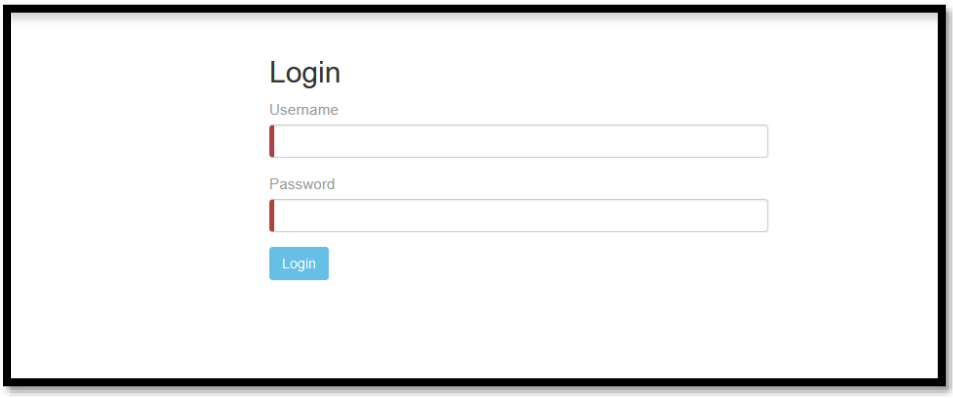

Figure 10: Gateway web interface log in page

The gateway dashboard appears as shown in Figure 11. This summarizes gateway connectivity, with more detailed configuration available from the toolbar at the top of the page. Details of each option can be described as follows.

- LAN – Configure Ethernet communications
- Wi-Fi – Configure W-iFi communications
- LoRa – Configure LoRa communications
- Settings – Gateway administration and management
- Logout – End the web interface session and return to the log in page

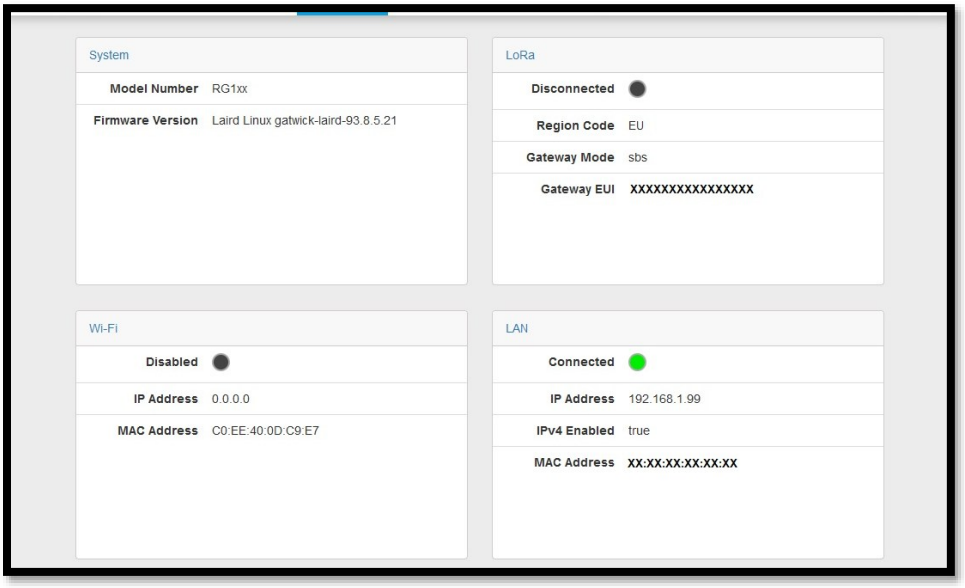

Figure 11: Gateway Dashboard page

7.1.4 Ethernet setup

The following describe the steps necessary to set the device up for Ethernet communications.

7.1.4.1 IPv4 Configuration

In the top menu, click **LAN.** Then click on **IPv4 Configuration** in the left submenu. This opens the page as shown in Figure 12.

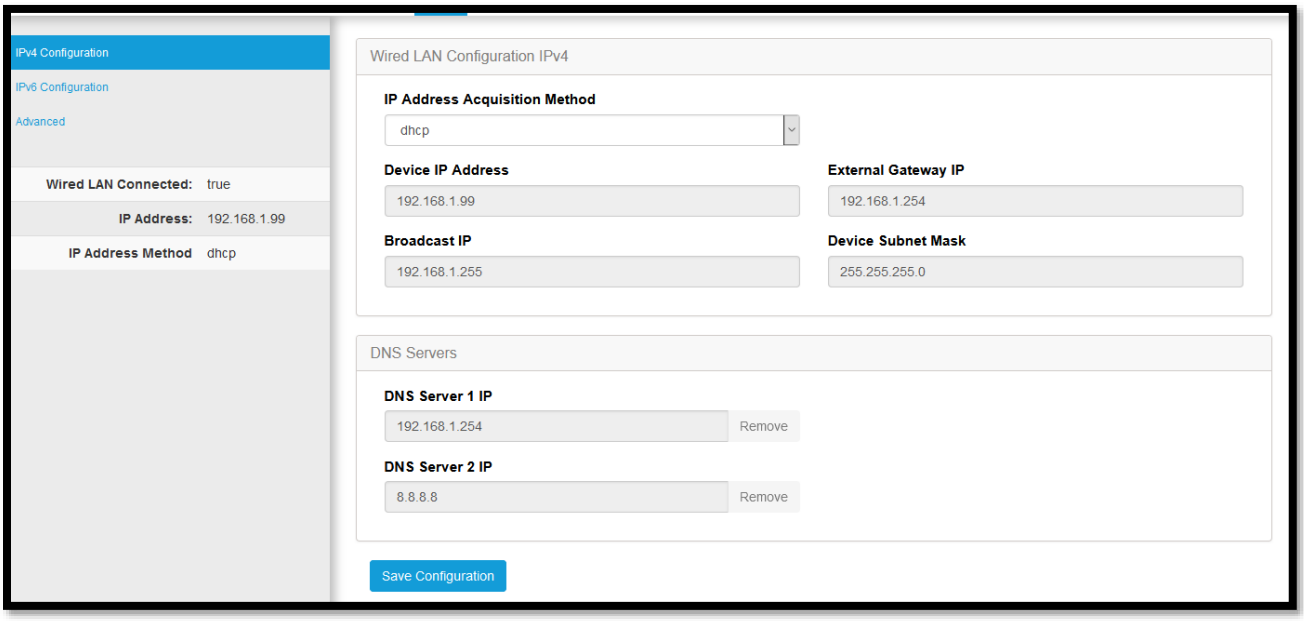

Figure 12: Gateway IPv4 configuration

The first page for configuring the Ethernet LAN connection is the IPv4 Configuration page. There are two basic modes of operation – DHCP and Static. These are selected in the IP Address Acquisition Method drop-down box. The gateway factory default setting is DHCP. The two settings can be described as follows.

- DHCP – When in DHCP mode, all settings are provided by the DHCP server. All configuration settings (except IP Address Acquisition Method) are greyed out. IP values provided by DHCP are displayed but cannot be changed
- Static – When the IP Address Acquisition Method is set to static, all IP settings are fixed and saved in the device. The external Gateway IP address is optional and may be left blank. DNS Server IP addresses are also optional. You may specify zero, one, or two DNS servers.

7.1.4.2 IPv6 Configuration

Click **LAN** in the top menu. Then click **IPv6 Configuration** in the left submenu. The IPv6 Configuration page appears as shown in Figure 13.

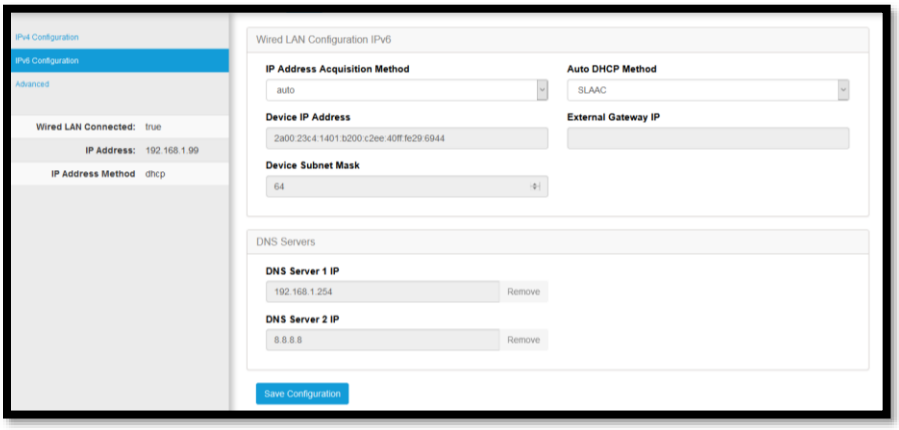

Figure 13: Gateway IPv6 configuration

The following modes are supported for IPv6 addressing.

- Static - When the IP Address Acquisition Method is set to static, all IP settings are fixed and saved in the device. As of June 2017, IPv6 static mode is only partially supported. Please see the software release notes for current information.
- DHCP – In DHCP mode, all settings are provided through communication with an IPv6 server on the network
- Auto – In auto mode, the auto DHCP method can be configured between Stateless or SLAAC

7.1.4.3 Advanced page

From the LAN page, clicking **Advanced** in the left submenu. IPv4 and IPv6 information appears as shown in Figure 14.

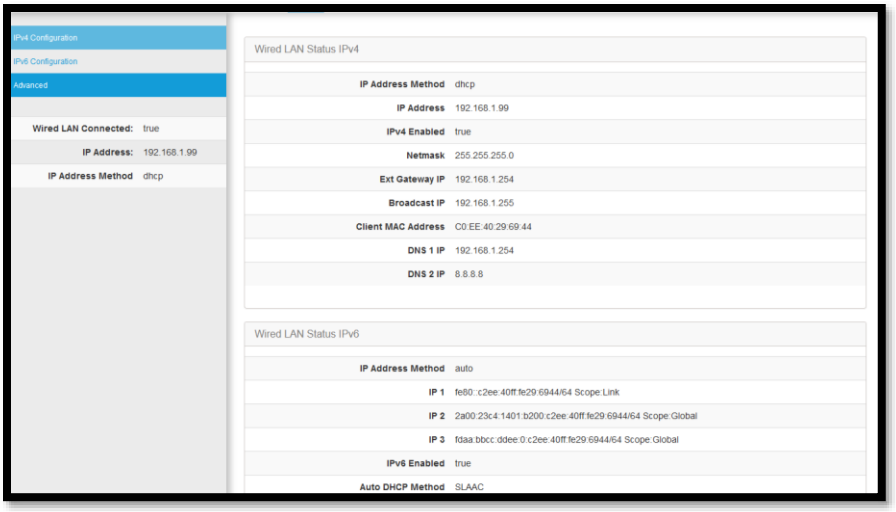

Figure 14: LAN Advanced page

7.1.5 Wi-Fi Setup

By default, the gateway's Wi-Fi radio is not configured to connect to a Wi-Fi network. The user must access the web interface on the gateway via the Ethernet interface to setup the Wi-Fi connection. This section describes the steps necessary.

7.1.5.1 Adding an access point

Click **Wi-Fi** in the top menu. The Wi-Fi page appears as shown in Figure 15.

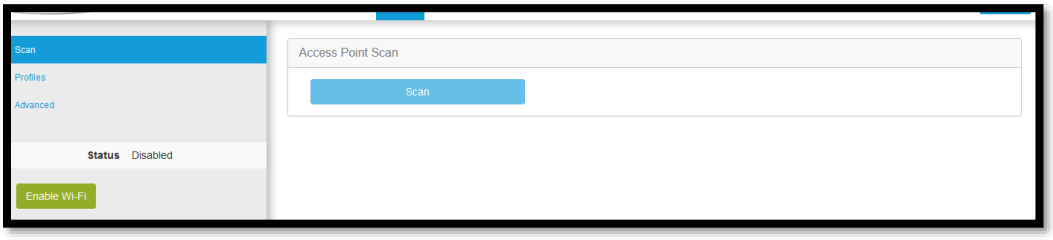

Figure 15: Web interface Wi-Fi page

Click **Enable Wi-Fi** to initialize the Wi-Fi hardware. The Wi-Fi LED on the gateway front panel flashes on and off, then Illuminates steadily. *Enable Wi-Fi* updates to display *Disable Wi-Fi* when Wi-Fi is active, as shown in Figure 16.

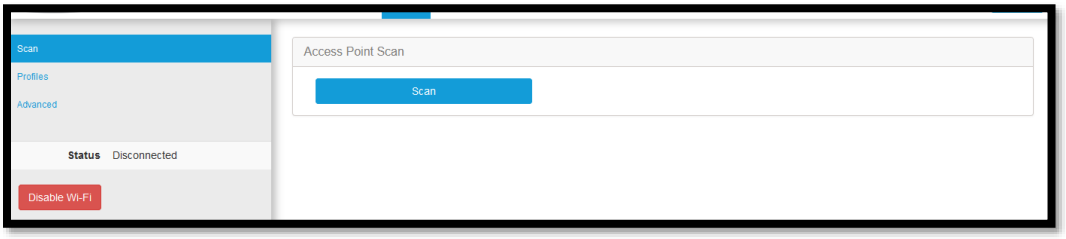

Figure 16: Web interface Wi-Fi page when Wi-Fi hardware is active

Click **Scan.** The gateway begins scanning for access points. The page displays results when complete, as shown in Figure 17.

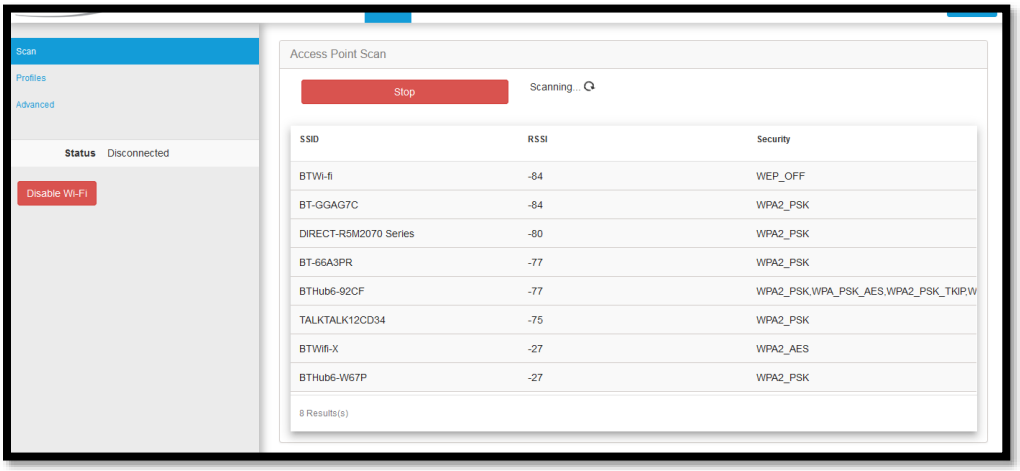

Figure 17: Access Point Scan results

Click your desired access point. Enter credentials as shown in Figure 18.

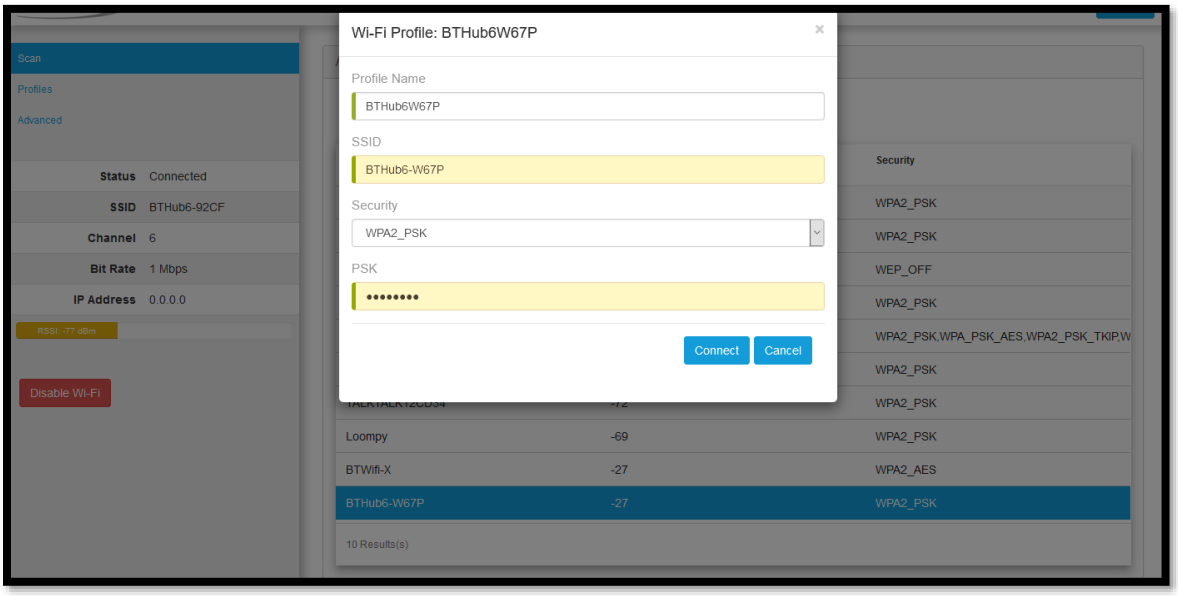

Figure 18: Access Point details

Click **Connect** to connect to the access point. Figure 19 shows the updated Wi-Fi page.

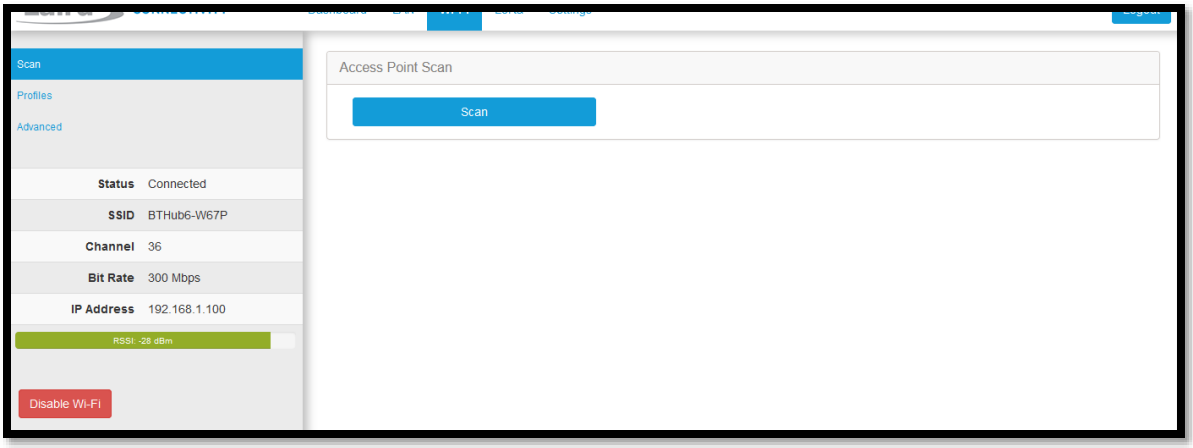

Figure 19: Successful connection to access point

7.1.5.2 Profiles page

Click **Profiles** in the left submenu of the Wi-Fi page. This page displays a summary of previously connected access points as shown in Figure 20.

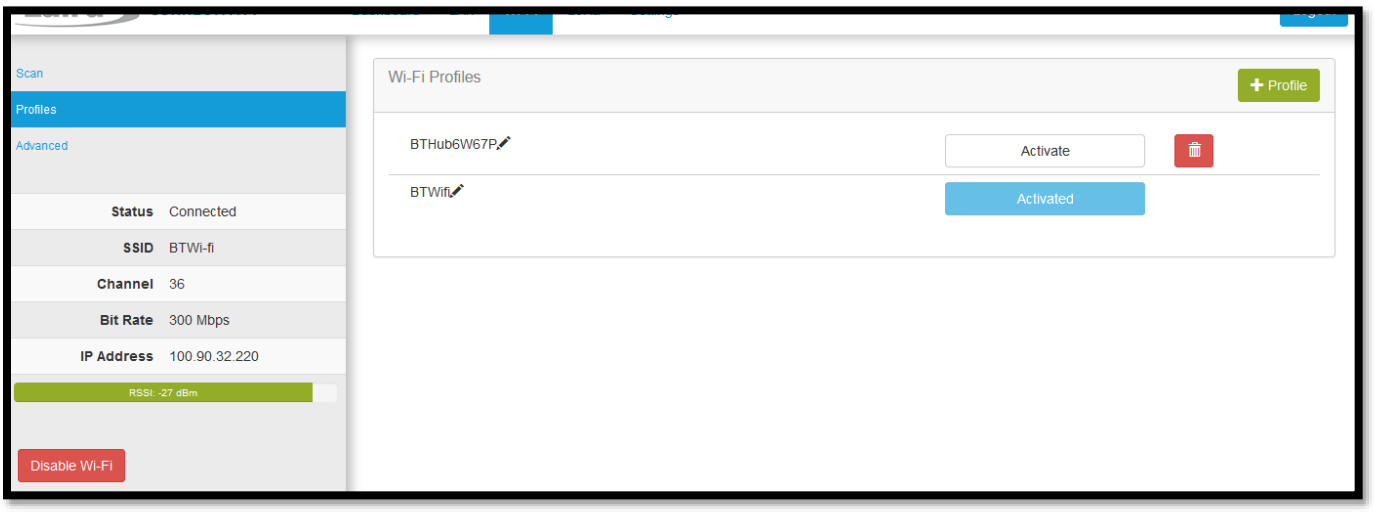

Figure 20: Wi-Fi Profiles page

This page allows you to modify settings for each, and to select the active access point.

7.1.5.3 Manually adding a profile

Manually add an access point by clicking **+ Profile** as shown in Figure 21. Then click **Add** to activate the new profile.

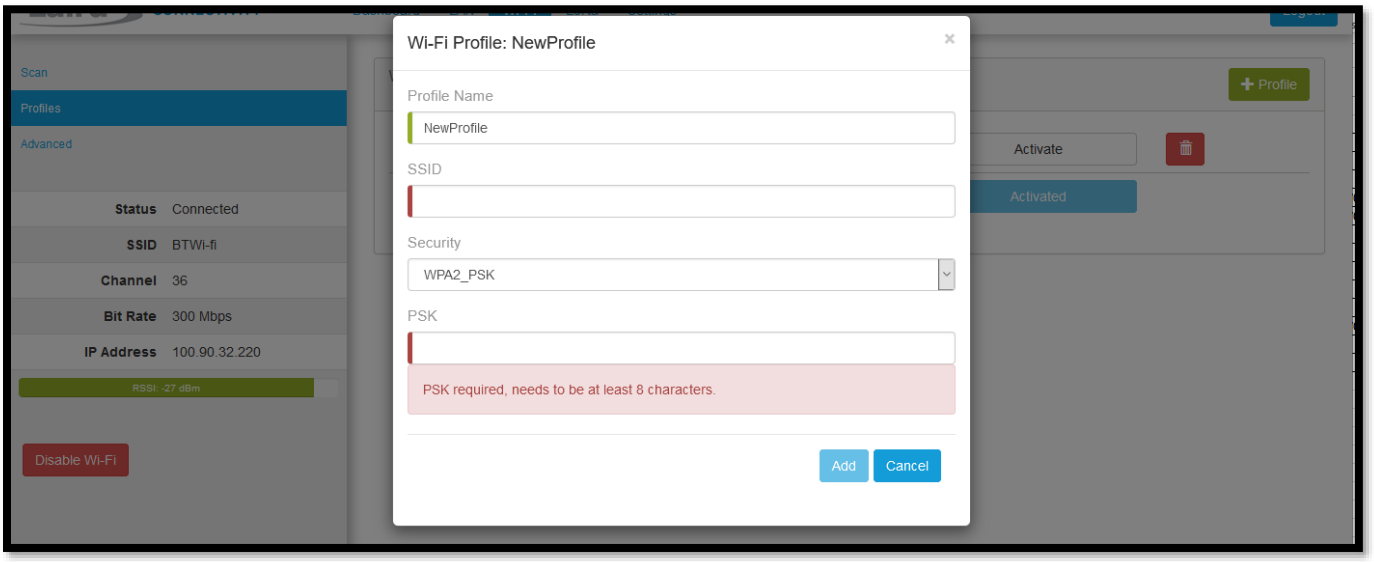

Figure 21: Manually adding an access point profile

7.1.5.4 Advanced page

Click **Advanced** in the submenu on the left to open Advanced page as shown in Figure 22. This page displays the parameters of the current access point.

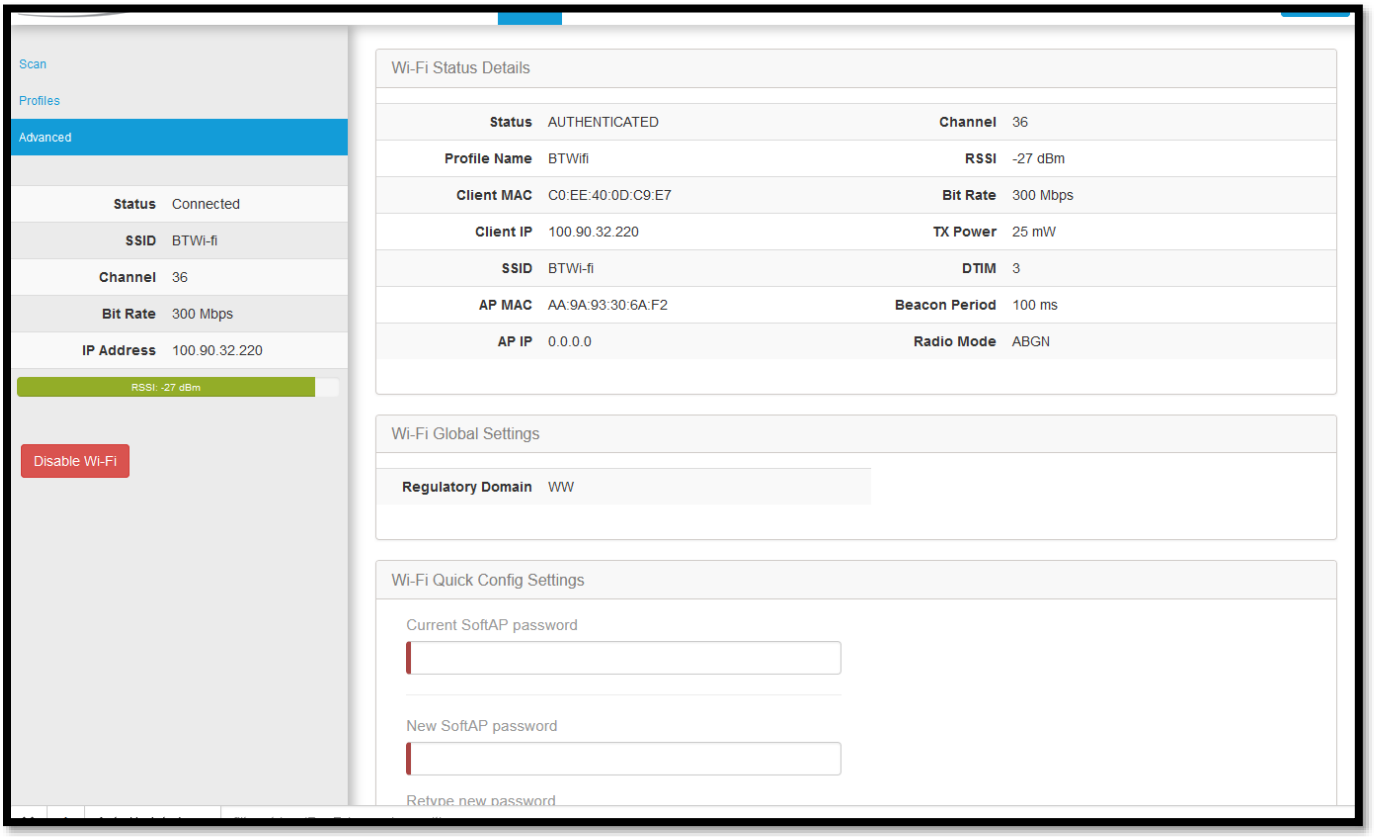

Figure 22: Wi-Fi Advanced page

7.2 Set Up Software

AWS IoT Core for LoRaWAN requires the usage of the Semtech BasicsStation Packet Forwarder v2.0.5. This is available in the RG1XX from firmware version 93.8.5.25 onwards. The firmware version on the gateway appears in the gateway dashboard as shown in Figure 23.

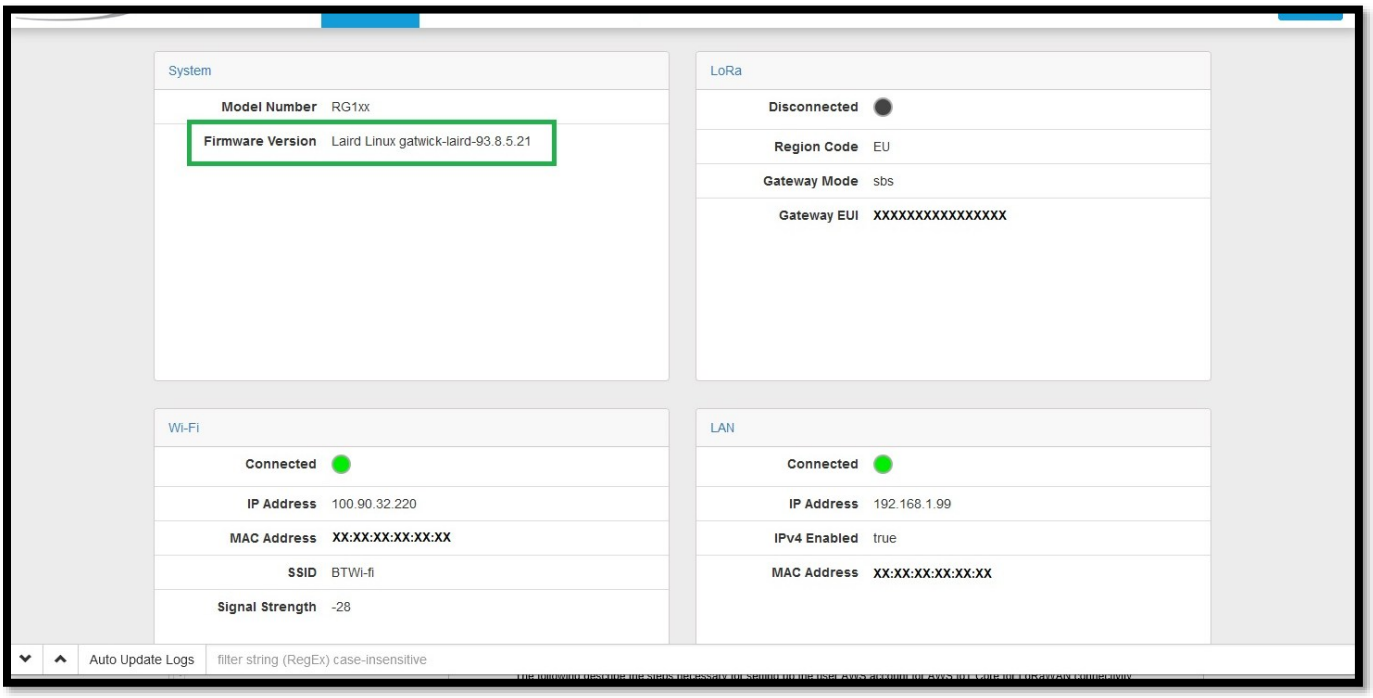

Figure 23: Verifying the gateway firmware version

If the firmware version is prior to 93.8.5.25, you must upgrade as shown in the Gateway OTA Updates section.

7.3 Configure the Gateway device

This section describes the activities performed on the gateway side to register it with AWS IoT Core for LoRaWAN. The gateway must be configured as described in the section Setup your AWS account and Permissions.

7.3.1 Enabling the Basics Station Packet Forwarder

Click **LoRa** in the main menu as shown in Figure 24.

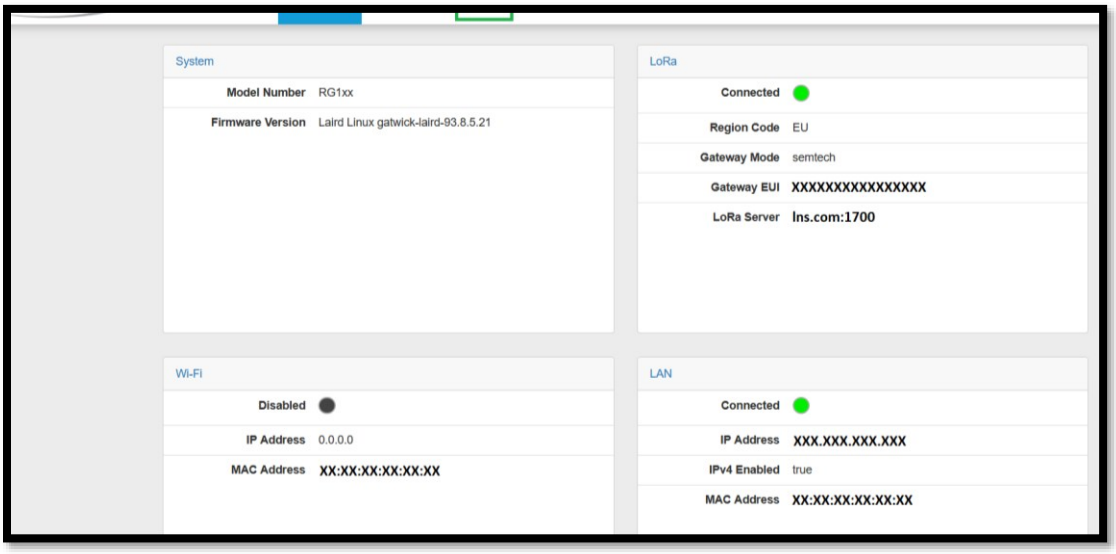

Figure 24: Opening the LoRa page from the gateway web interface

Click **Forwarder** in the left submenu. Set the *Mode* dropdown to Semtech Basics Station as shown in Figure 25.

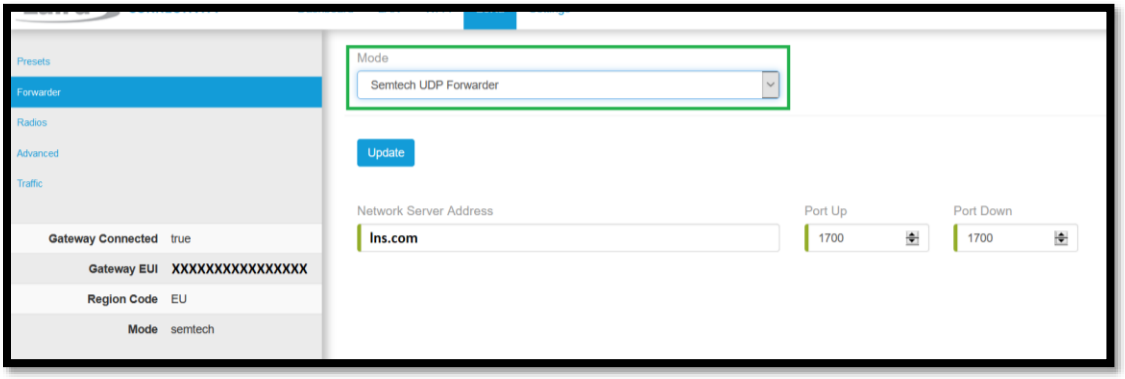

Figure 25: Packet Forwarder selection

7.3.2 Configuring end points

The Basics Station configuration page appears as shown in Figure 26. Configure details of the CUPS and LRC endpoints in the *Server Configuration* group.

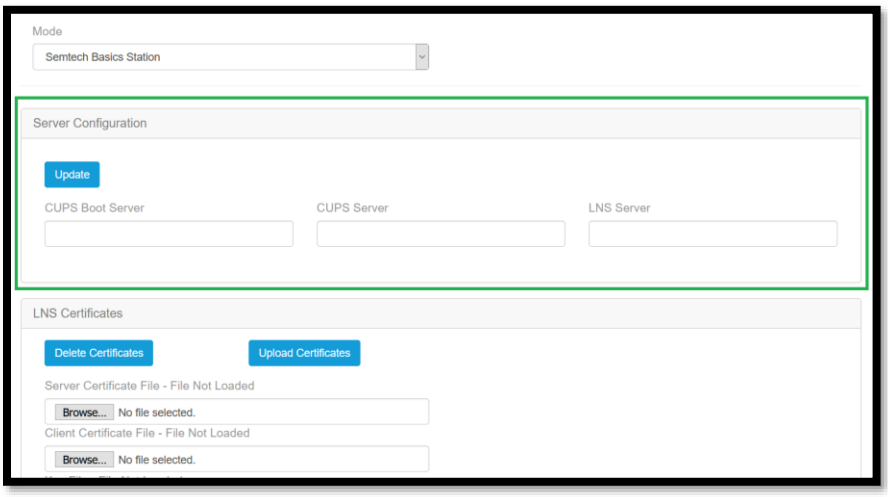

Figure 26: Basics Station Server Configuration

- The 'CUPS Server' and 'CUPS Boot Server' should be set to the CUPS Endpoint value noted during section 6.3.
- The 'LNS Server' should be set to the LNS Endpoint value noted during section 6.3.

Click **Update** to store the values in the gateway.

7.3.3 Configuring LNS certificates

Add Certificate data for the LNS aspect of the AWS IoT Core for LoRaWAN to the gateway via the *LNS Certificates* group as shown in Figure 27.

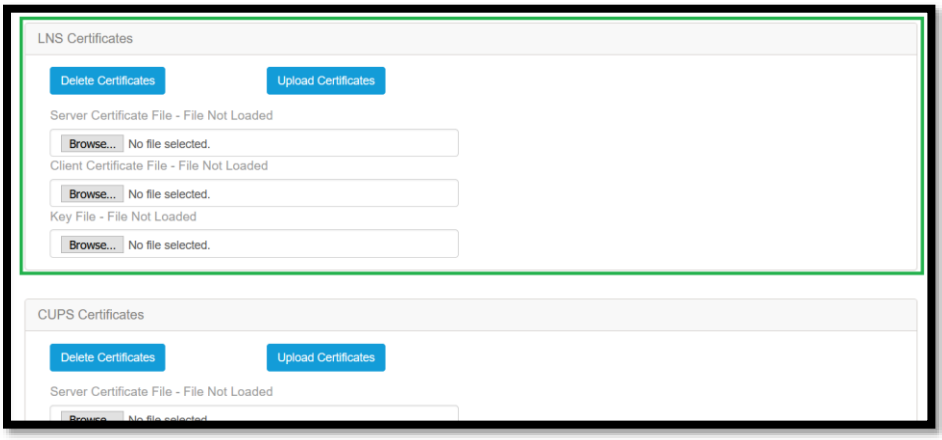

Figure 27: LNS Certificates group

- 'Server Certificate File' is the LNS Trust Certificate file stored during the steps described in section 6. This file is named **lns.trust**. Select *All Files (*.*)* in the file browse dialog to make the file visible.
- 'Client Certificate File' is the Gateway Certificate file stored during the steps described in section 6. This has the ***.pem** extension. Select *All Files (*.*)* in the file browse dialog to make the file visible.
- 'Key File' is the Gateway Private Key file stored during the steps described in section 6. This file has the ***.key** extension.

In all cases, click **Browse** to navigate to the file location on the web interface client machine. Click **Upload Certificates** to upload the files to the gateway.

7.3.4 Configuring CUPS certificates

Add details of the CUPS server, if required, via the CUPS Certificates group, as shown in Figure 28.

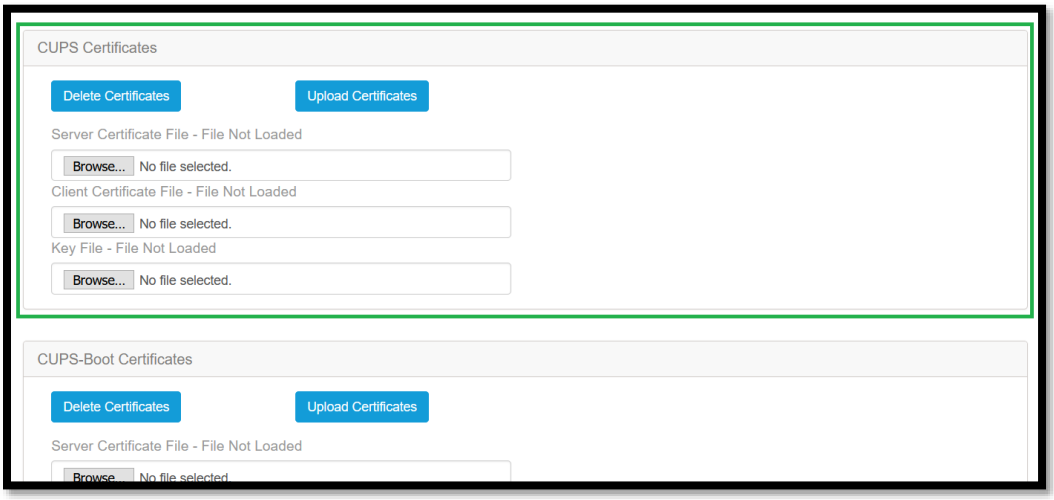

Figure 28: CUPS Certificates group

- The 'Server Certificate File' is the CUPS Trust Certificate file stored during the steps described in section 6. This has the filename **cups.trust**. Select *All Files (*.*)* in the file browse dialog to make the file visible.
- 'Client Certificate File' is the Gateway Certificate file stored during the steps described in section 6. This has the ***.pem** extension. Select *All Files (*.*)* in the file browse dialog to make the file visible.
- 'Key File' is the Gateway Private Key file stored during the steps described in section 6. This has the ***.key** extension.

Click **Upload Certificates** after you select all files to transfer the files to the gateway.

7.3.5 Configuring CUPS Boot certificates

Add details of the CUPS Boot server, if required, via the *CUPS-Boot Certificates* group, as shown in Figure 29.

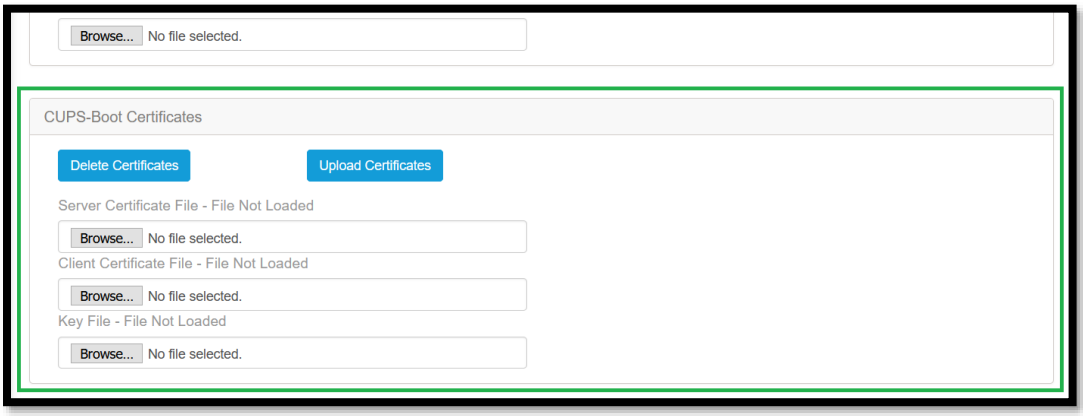

Figure 29: CUPS Boot Certificates group

- The 'Server Certificate File' is the CUPS Trust Certificate file stored during the steps described in section 6. This has the filename **cups.trust**. Select *All Files (*.*)* in the file browse dialog to make the file visible.
- 'Client Certificate File' is the Gateway Certificate file stored during the steps described in section 6. This has the ***.pem** extension. Select *All Files (*.*)* in the file browse dialog to make the file visible.
- 'Key File' is the Gateway Private Key file stored during the steps described in section 6. This has the ***.key** extension.

Click **Upload Certificates** after selecting all files to transfer the files to the gateway.

7.3.6 Finalising gateway configuration

Once you have entered the certificate and endpoint data, reboot the RG1xx to allow the changes to take effect. To reboot, click **Settings** in the top menu, and then **Reboot** in the left submenu as shown in Figure 30. The gateway will restart within a minute, then establish communication with the AWS IoT Core for LoRaWAN instance.

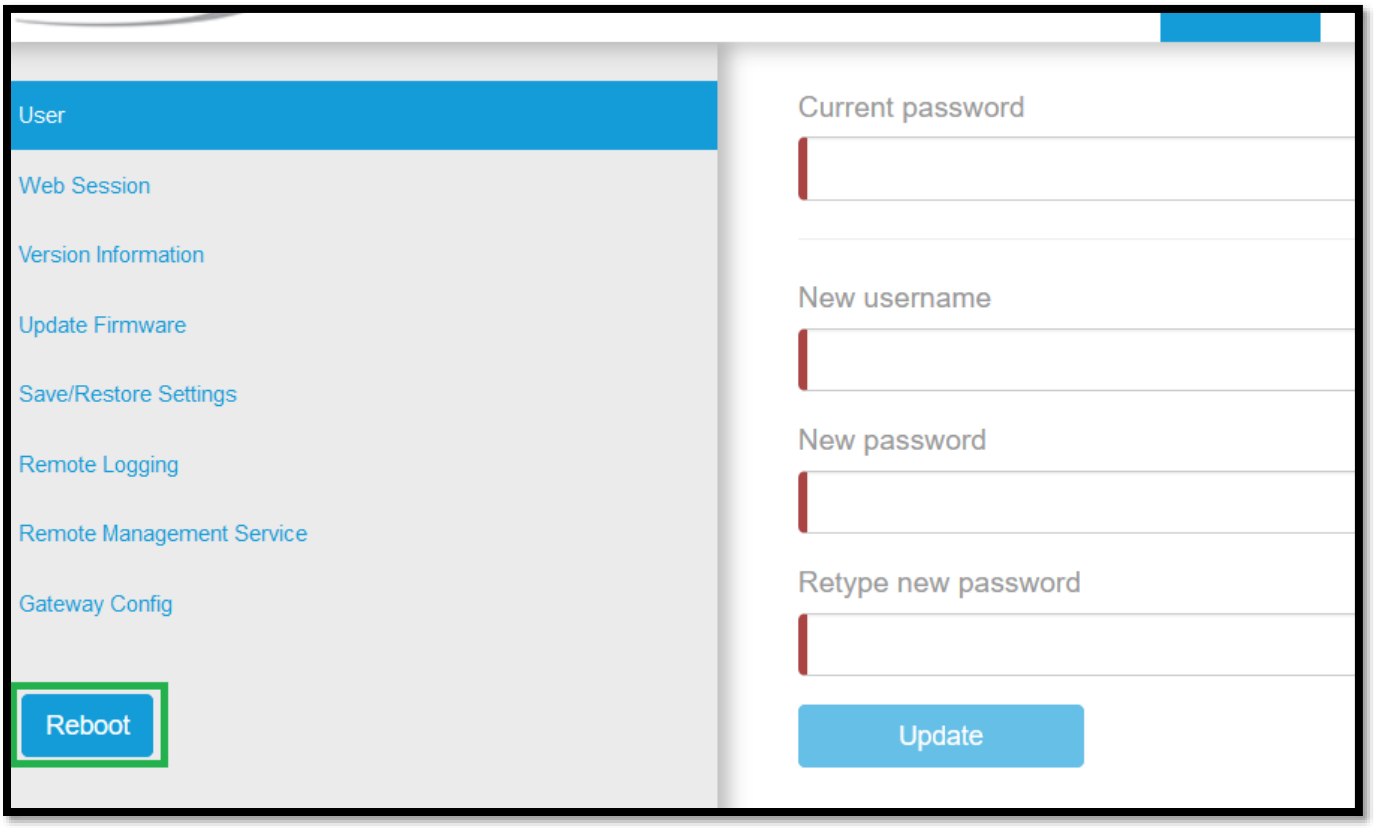

Figure 30: Rebooting the RG1xx

8 Set Up the Sensor

The following describes the server side steps necessary for integrating RS1XX sensors with AWS IoT Core for LoRaWAN.

8.1 Set up hardware

The following describe the steps required to setup an RS1XX sensor. Figure 31 shows the hardware features of the sensor.

1. Temperature and humidity sensor
2. Bluetooth button
3. LEDs
4. Fixing holes

Figure 31: Sentrius RS1XX Sensor hardware features

8.1.1 Physical Connectivity

Insert the batteries supplied with the sensor into the sensor battery port, as shown in Figure 32, then close the port with the included screw. You may also use lithium batteries, which are suitable for applications where frequent uplinks are expected.

Figure 32: Sentrius sensor battery port

8.1.2 Sensor LEDs

The LED array visible on the sensor housing is shown in Figure 33. Table 2 describes the purpose of each LED.

Figure 33: RS1XX LEDs

Table 2: RS1XX LEDs

Label	Purpose
1	Flashes orange when a network is being searched for. Flashes green following successful connection to a network.
2	Flashes blue when BLE communication is in progress.

8.2 Set Up Software

8.2.1 Determining the Sensor EUI

The EUI of each sensor to be registered with AWS IoT Core for LoRaWAN is required when the sensor is registered. The EUI is listed on the label on the rear of the sensor, as shown in Figure 34, or via the mobile app on the LoRa Settings screen as shown in Figure 35.

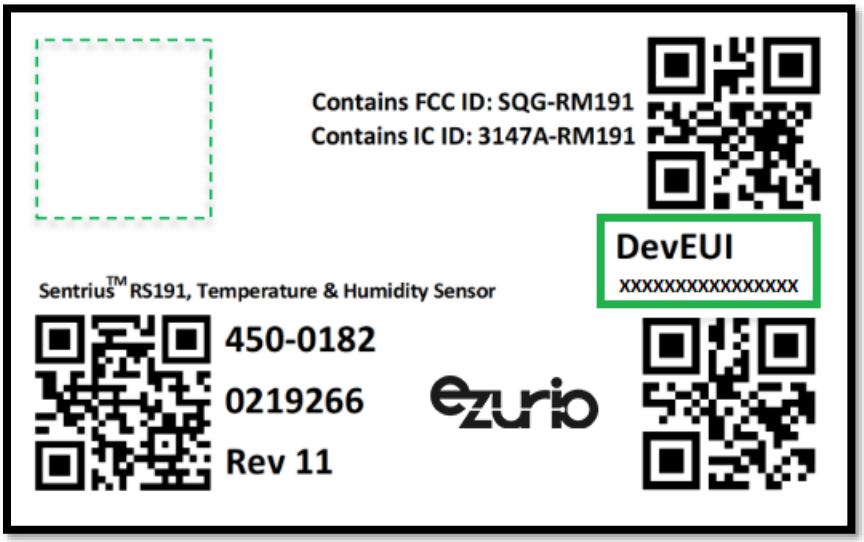

Figure 34: Determining the sensor EUI from housing label

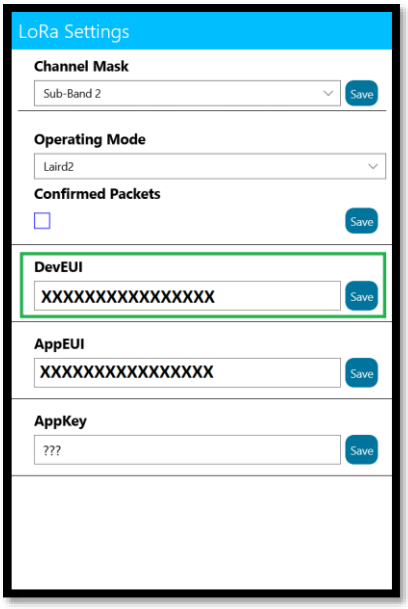

Figure 35: Determining the sensor EUI from the mobile app

8.2.2 Determining the App EUI

Every sensor within an application must be defined by a 64-bit (16-character hexadecimal string) value. Follow the guidance defined in 'Join EUI Guidance' in reference [S] when defining the value. View and modify the App EUI via the mobile app 'LoRa Settings' page as shown in Figure 36.

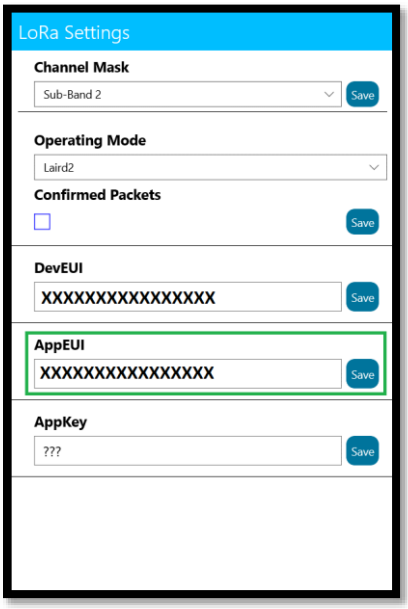

Figure 36: Join EUI configuration via mobile app

8.2.3 Determining the App Key

Every sensor within an application must be assigned a 128-bit (32-character hexadecimal string) app key value. Follow the 'Security Keys Guidance' defined at reference [T] when defining the key value. View and modify the app key via the mobile app on the LoRa Settings page, as shown in Figure 37. Alternatively, you may use the App Key provided on the external label on the rear of the sensor.

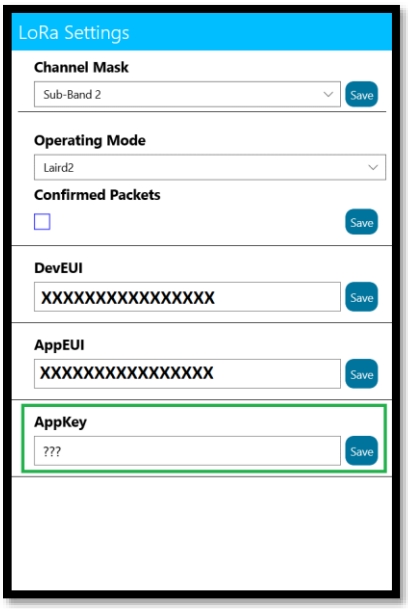

Figure 37: Configuring the AppKey via the mobile app

8.2.4 Determining the Sensor Region

You must provide region details when registering the sensor with AWS IoT Core for LoRaWAN. They are found on the Update Firmware page as shown in Figure 38.

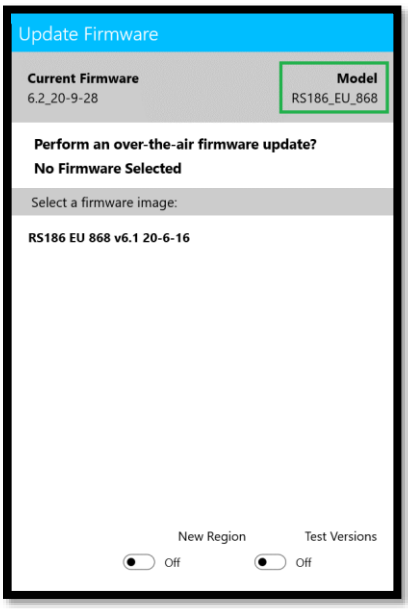

Figure 38: Determining the sensor region via the mobile app

8.2.5 Determining the Sensor LoRaWAN Specification Version

All RS1XX sensors currently support v1.0.2 of the LoRaWAN Specification.

8.2.6 Determining the Sensor Activation Type

RS1XX sensors support only the OTAA type of activation.

8.2.7 Determining the Sensor Regional Parameters Revision

All RS1XX sensors currently implement v1.0.2B of the LoRaWAN Regional Parameters specification.

9 Application Examples

The following describe some applications to test the sensor connectivity and demonstrate AWS features.

9.1 Ezurio Protocol Format example

Before implementing this example application, set the sensor Packet Format to 'Laird' or 'Laird 2'. Refer to reference [O] for further details of configuring the Packet Format, and details of the available Packet Formats.

Lambda code in NodeJS 10.x format is provided on our GitHub page [W] for decoding the Ezurio format payload data into meaningful values. Further details of the protocol implemented by the Laird and Laird 2 Packet Format are provided in the RS1xx Protocol Description [U].

The architecture of the application is shown in Figure 39. Messages received from the sensor are passed to the 'Decoder' Rule. This invokes the 'Decoder' Lambda function, which extracts payload data from the messages and converts into human readable data. Output from the Lambda function is published to the 'Decoded' MQTT topic, where the data can be inspected via AWS' MQTT Client.

A second Rule, 'Extractor', subscribes to the 'Decoded' Topic and extracts timestamp, temperature and DevEUI data. These are published to a second MQTT topic, 'Extracted', and stored in the 'Extracted' Dynamo Database table for later use.

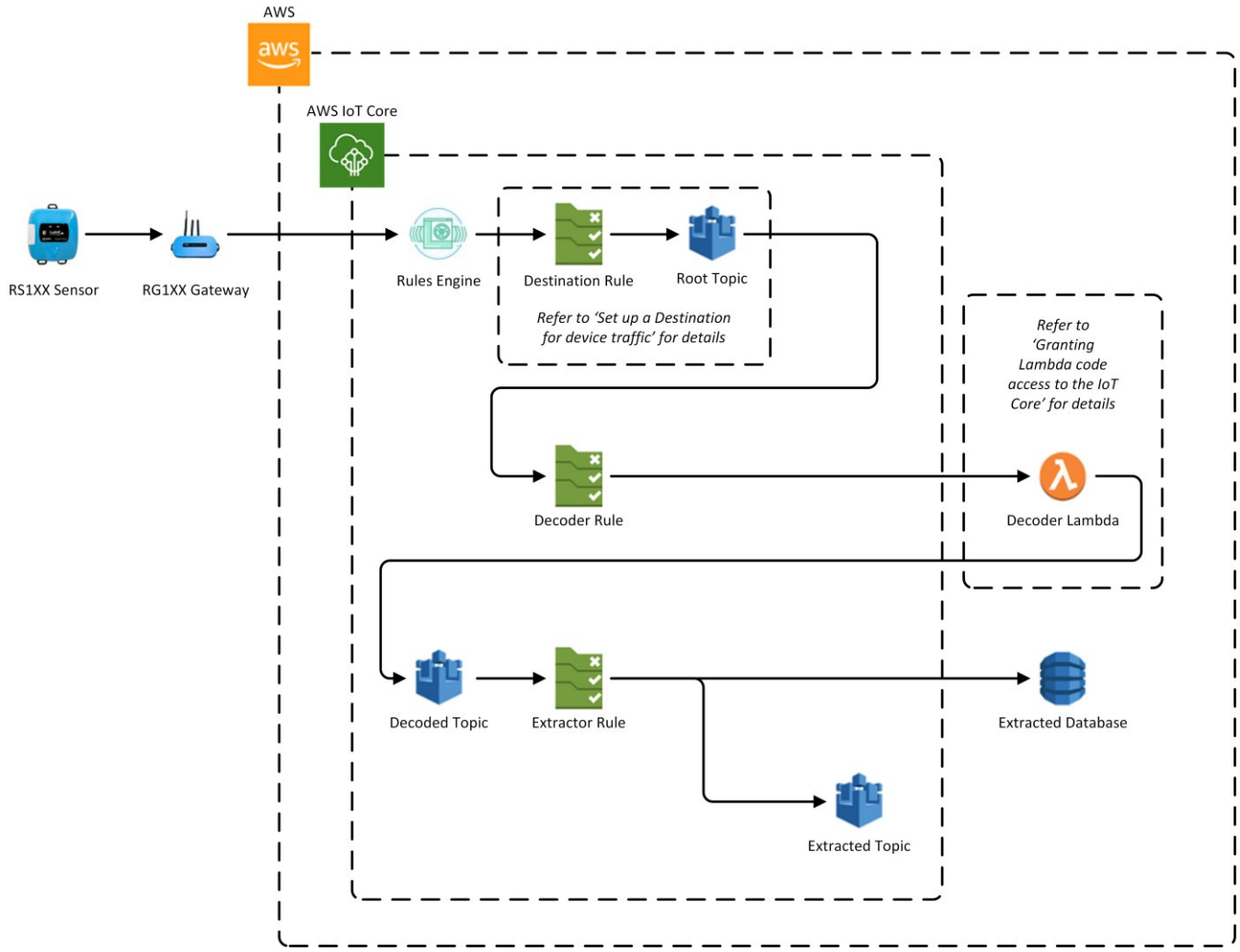

Figure 39: Ezurio Packet Format application architecture

9.1.1 Creating the Decoder Lambda function

First, create the Decoder Lambda function. As shown in Figure 40, click **Lambda** from the AWS landing page main menu.

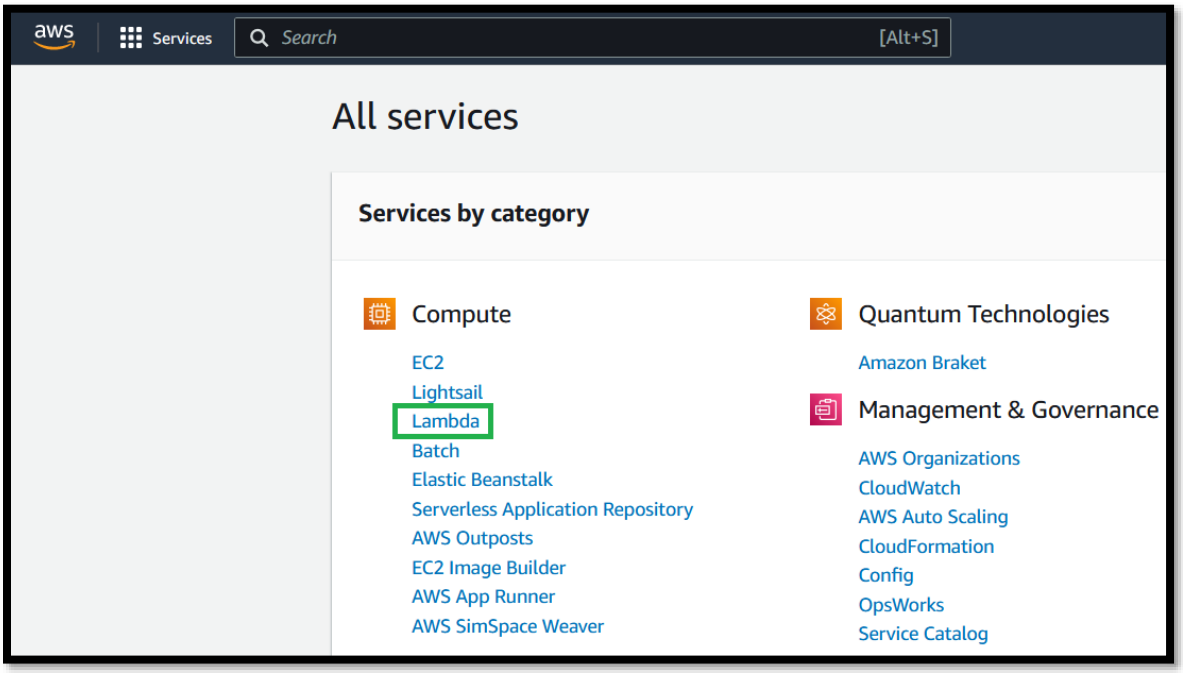

Figure 40: Creating a new Lambda function

This opens the main Lambda page as shown in Figure 41. Click **Create function**.

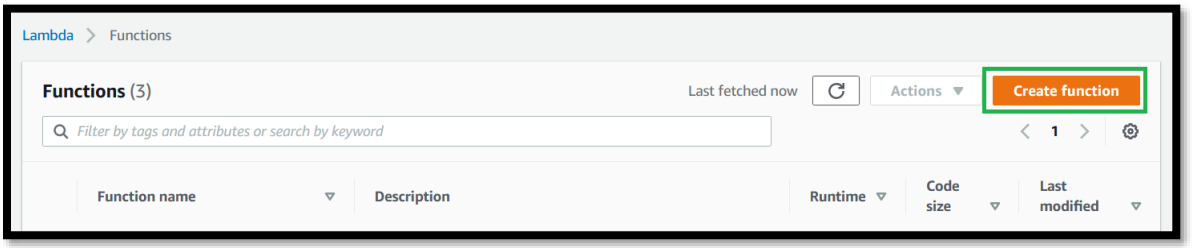

Figure 41: Main Lambda page

This opens the page shown in Figure 42. *Author from scratch* should be selected. Set *Function name* to "Decoder' and *Runtime* to "NodeJS 16.x". Click **Create function**.

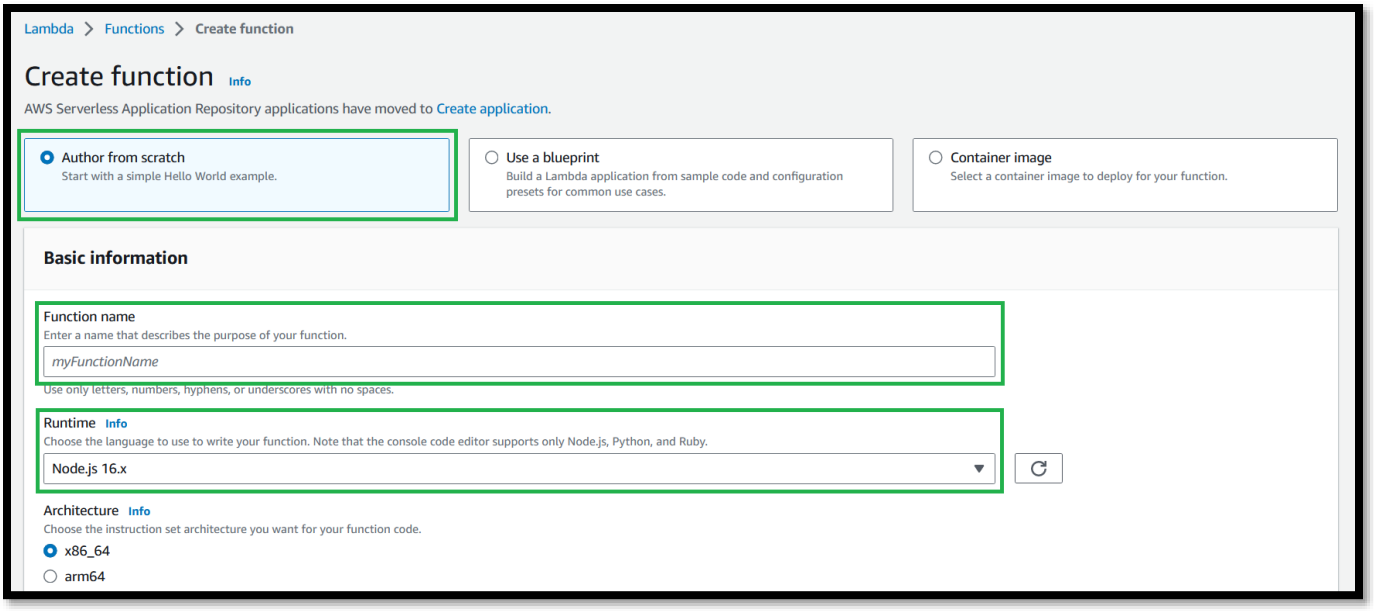

Figure 42: Setting Lambda function information

The Lambda function designer is as shown in Figure 43. Note the ARN of the Lambda for later use. It is passed to the query used in the Decoder Rule.

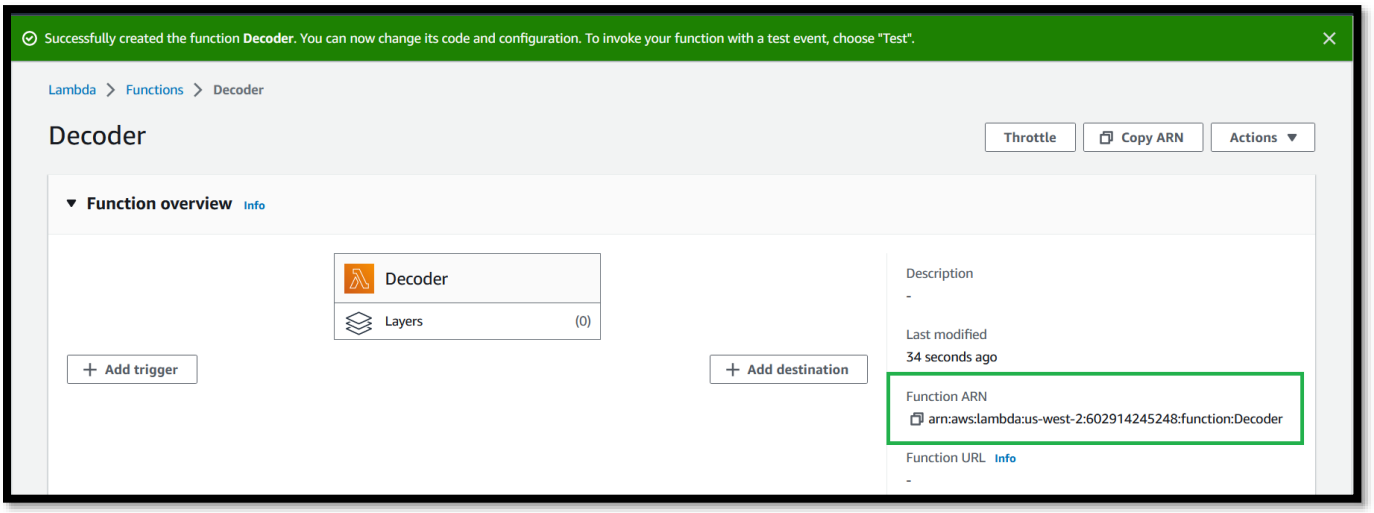

Figure 43: Lambda function designer

Scroll down to the Lambda function code, as shown in Figure 44. The example file, index.js, should be deleted. Add the Decoder files either by creating files “library_laird.js”, “index.js” and “messages_laird.js” and copying/pasting the content, or by creating a zip file containing the files and uploading to AWS. Note you must manually create the .zip and it must contain the three files needed in the root directory of the archive.

Figure 44: Lambda function body

Figure 45 shows the method where a zip file is uploaded with the example code incorporated in a zip file.

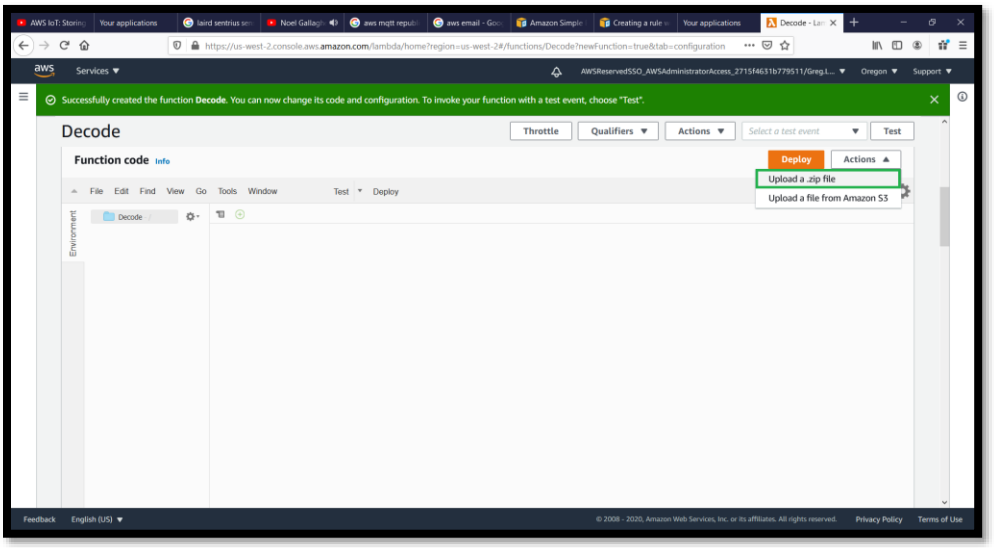

Figure 45: Uploading Lambda function content

Once the Lambda code is available, the function content appears as shown in Figure 46.

Note: If the Lambda files are added via a zip file, they are automatically deployed. If manually added or edited, **Deploy** must be clicked to update their content. Refer to Figure 46.

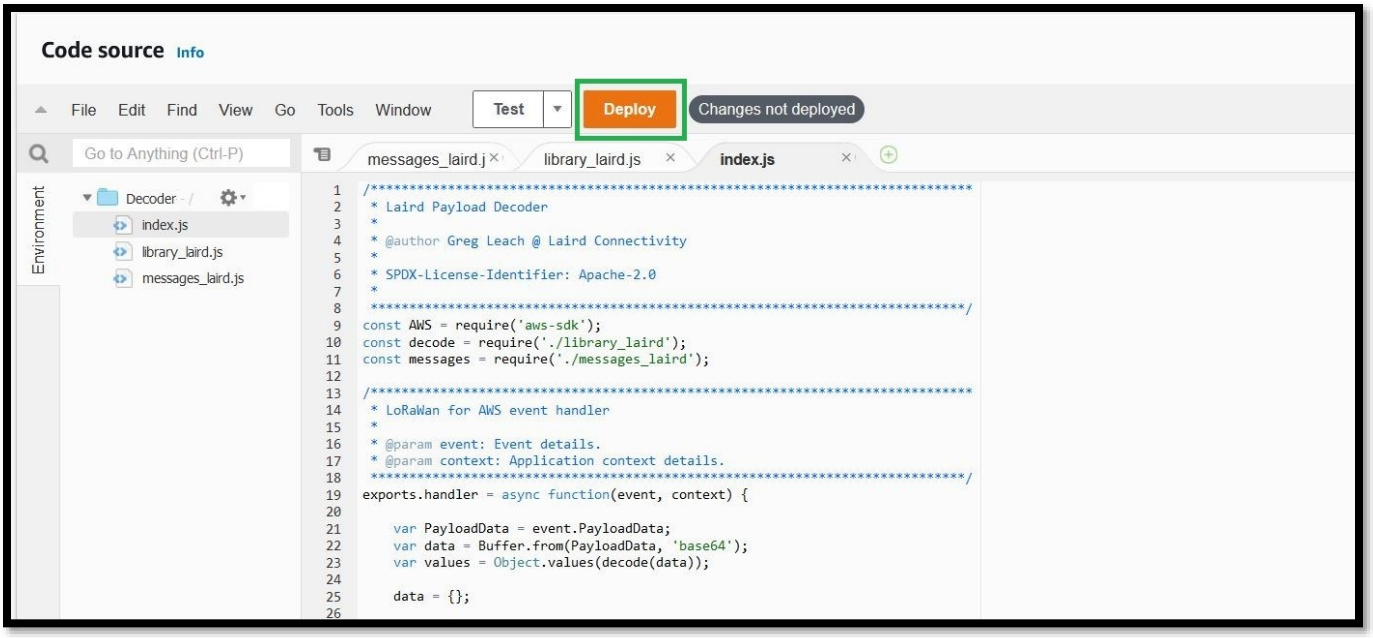

Figure 46: Completed Lambda function code

9.1.2 Creating the Decoder Rule and Decoded topic

You can now invoke the Lambda code from a Rule. The output of the Lambda is published to an MQTT topic, 'Decoded', for later inspection.

From the AWS main menu, click **IoT Core** as shown in Figure 47.

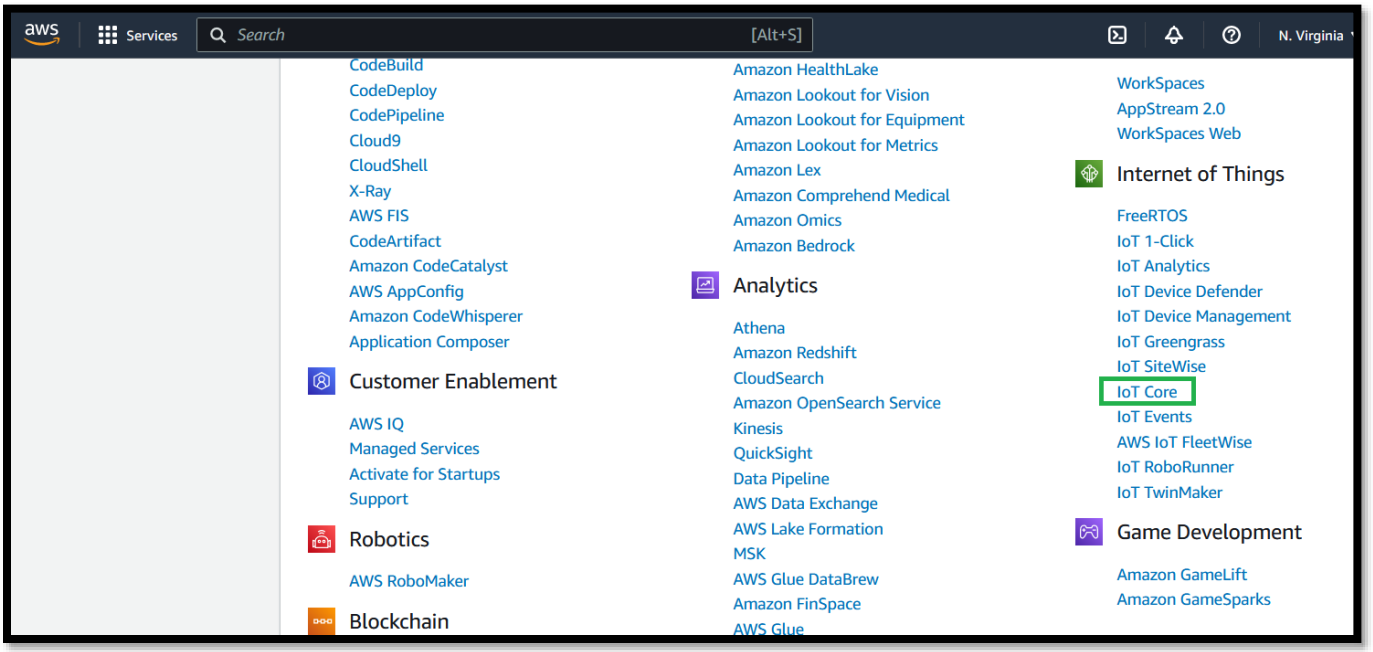

Figure 47: Opening AWS IoT Core

From the side menu, click **Message Routing**, then click **Rules**, as shown in Figure 48. This opens the AWS IoT Core Rules Engine.

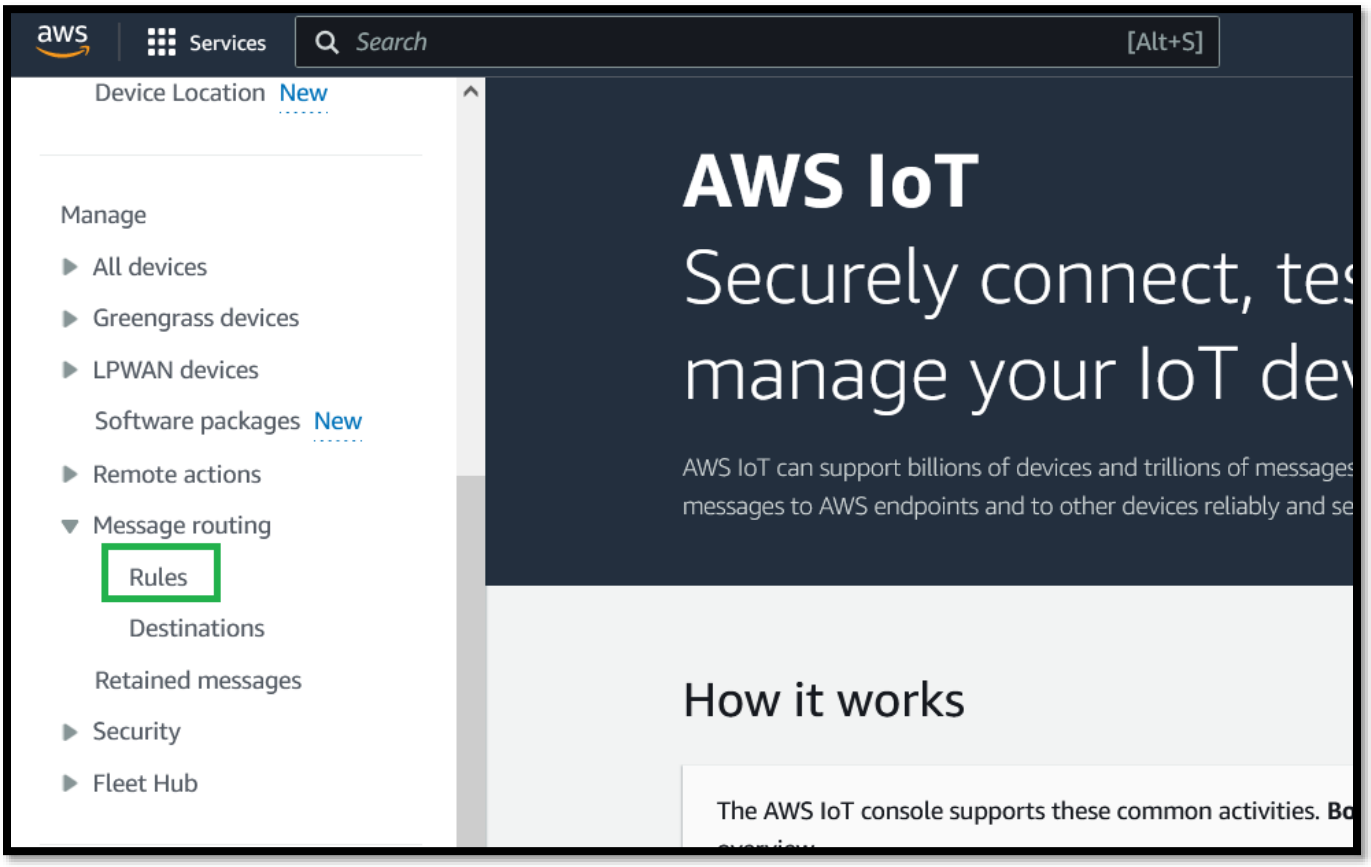

Figure 48: Opening IoT Core Rules Engine

Click **Create** as shown in Figure 49 to create a new Rule.

Figure 49: Creating a new Rule

Enter "Decoder" for the rule name as shown in Figure 50, then click **Next**.

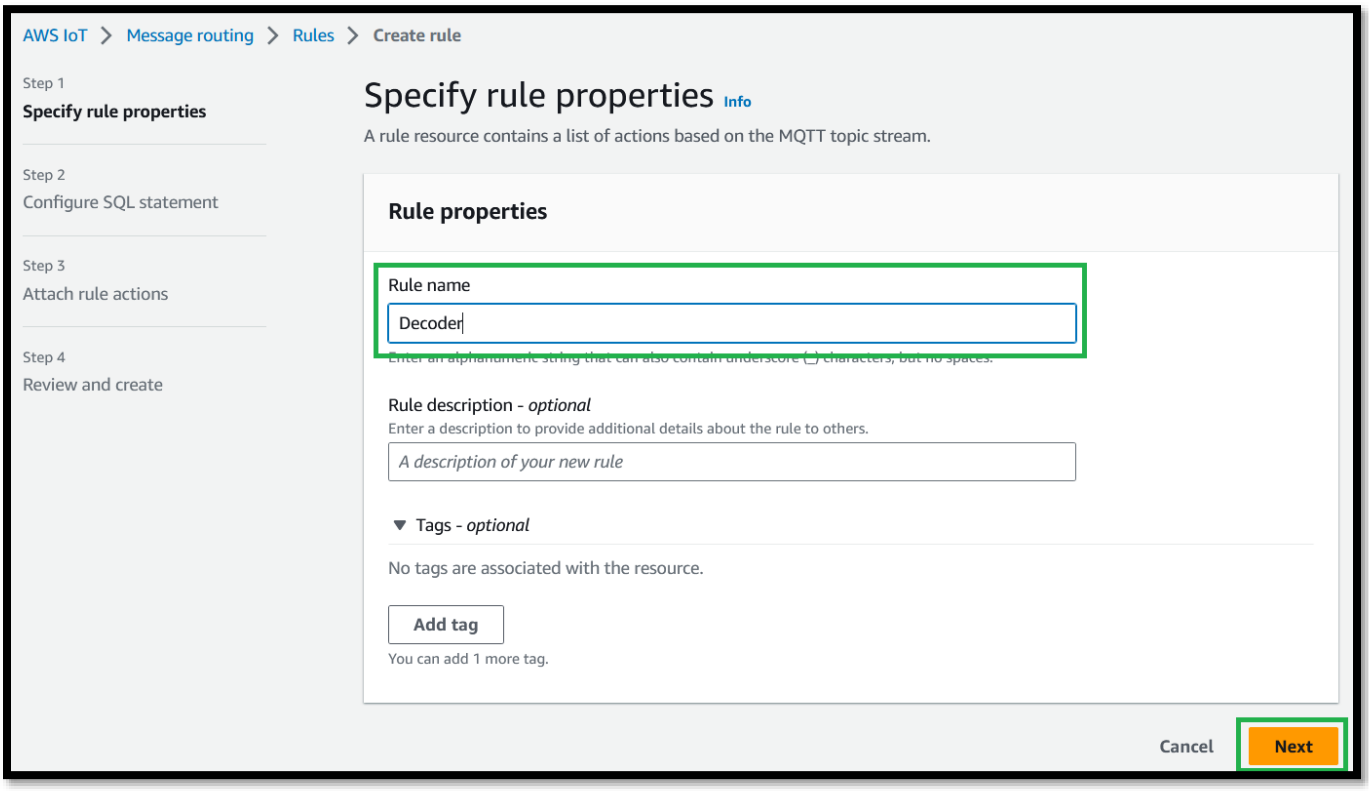

Figure 50: Creating the Decoder Rule

The Configure SQL statement step appears as shown in **Figure 51**.

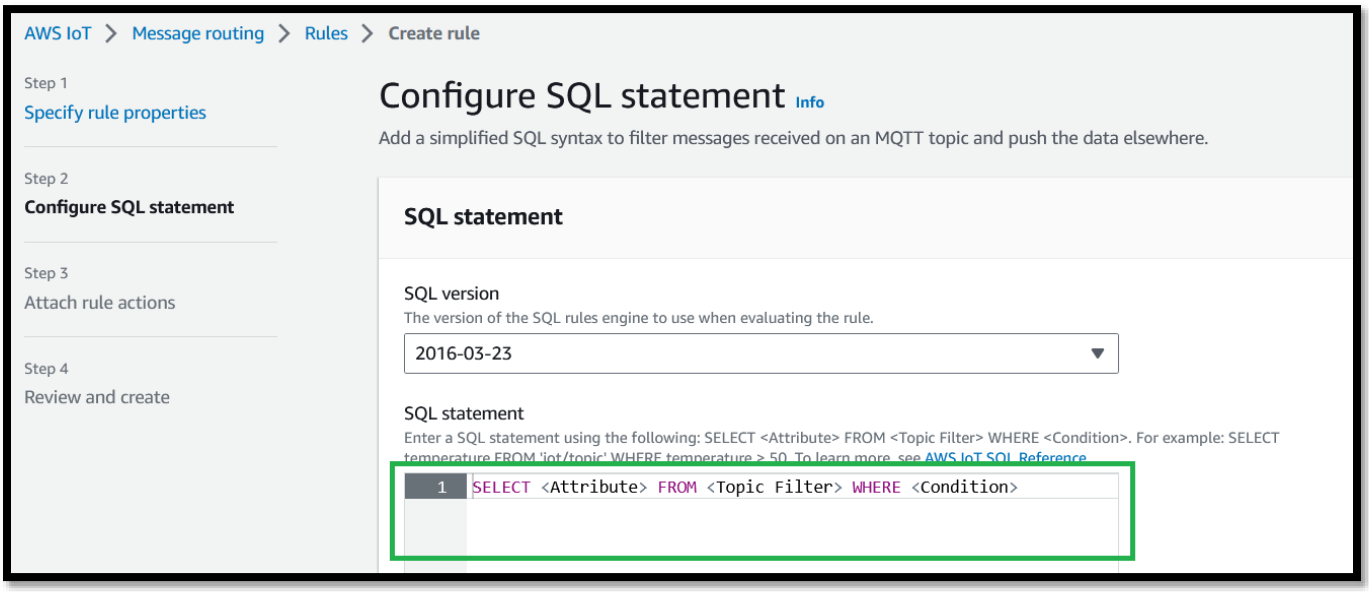

Figure 51: Configuring Rule SQL statement

This is where data is extracted from the root topic for use elsewhere. Set the query statement as follows:

```
SELECT aws_lambda("Decoder ARN", *) as output FROM 'Root MQTT Topic'
```

Replace "Decoder ARN" with the Lambda ARN noted earlier, within the double quotes. Refer to Section 6.4.3 for details of the Root MQTT Topic, this is enclosed within apostrophes.

Refer to Figure 52 for the expected formatting and appearance.

Figure 52: Rule query statement

Click **Next** to proceed to the **Attach rule actions** step, as shown in Figure 53. During this step, an Action is added to publish output of the Lambda to an MQTT topic. **Action 1** should be clicked to select the publish action.

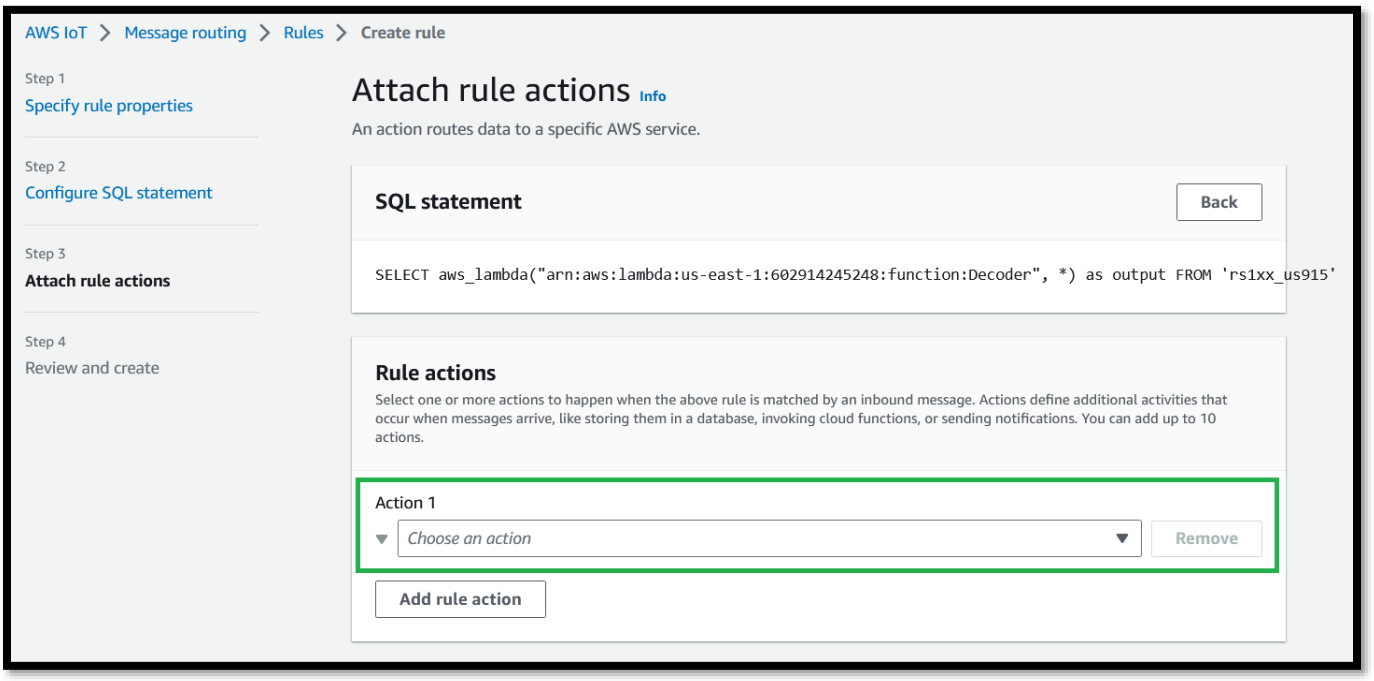

Figure 53: Adding the publish Action

Select **Republish a message to an AWS IoT topic** as shown in Figure 54. The MQTT topic where output of the Lambda should be republished should also be set here in the **Topic** field.

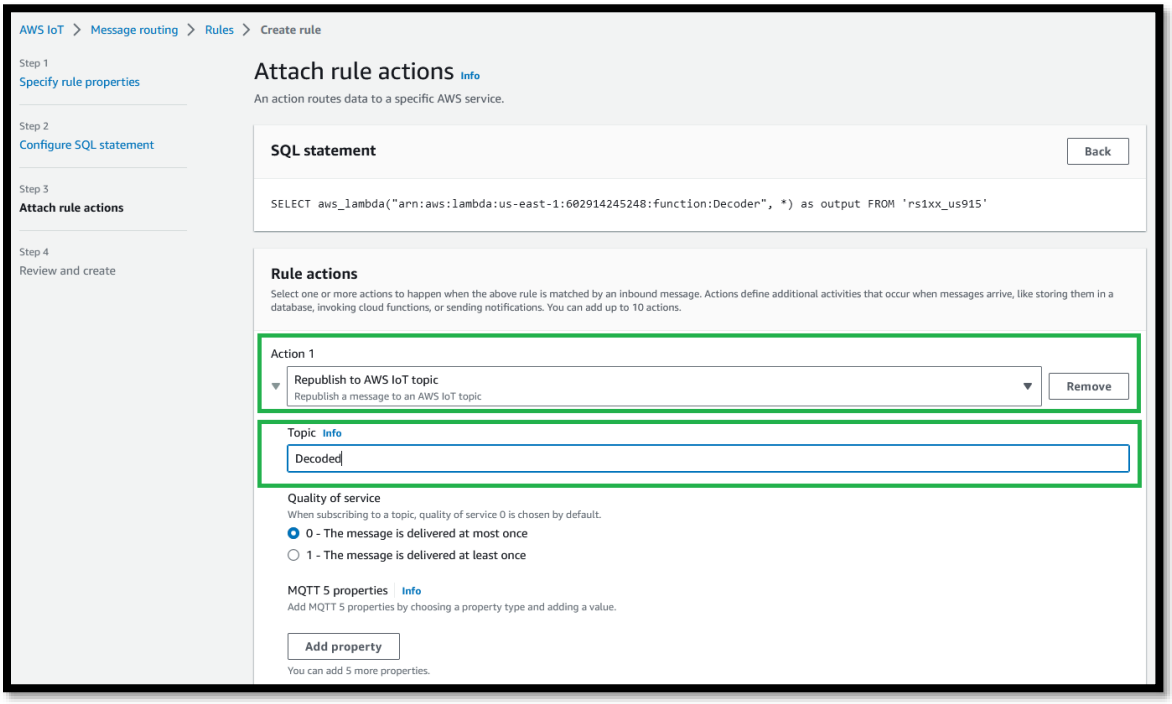

Figure 54: Selecting the Republish Action

Scroll down and click **Create new role** as shown in Figure 55.

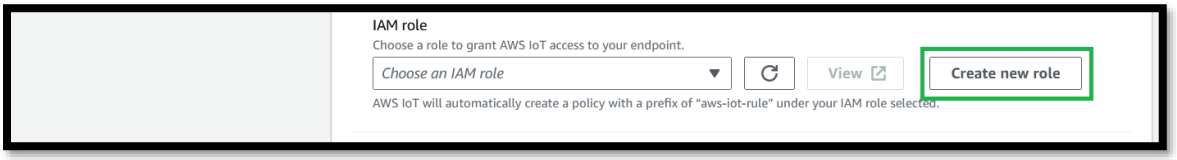

Figure 55: Configuring the Decoder IAM role

Set the name of the Role as Decoder_Role and click Create as shown in Figure 56.

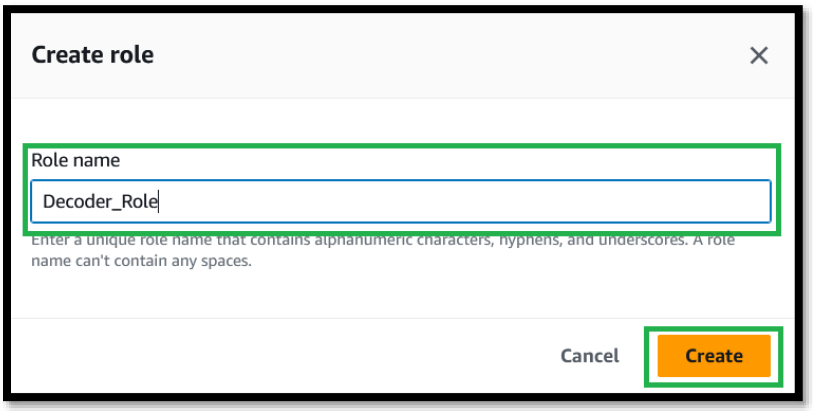

Figure 56: Adding Decoder role details

A second Action must now be added to the Rule. As shown in Figure 57, click **Add rule action**.

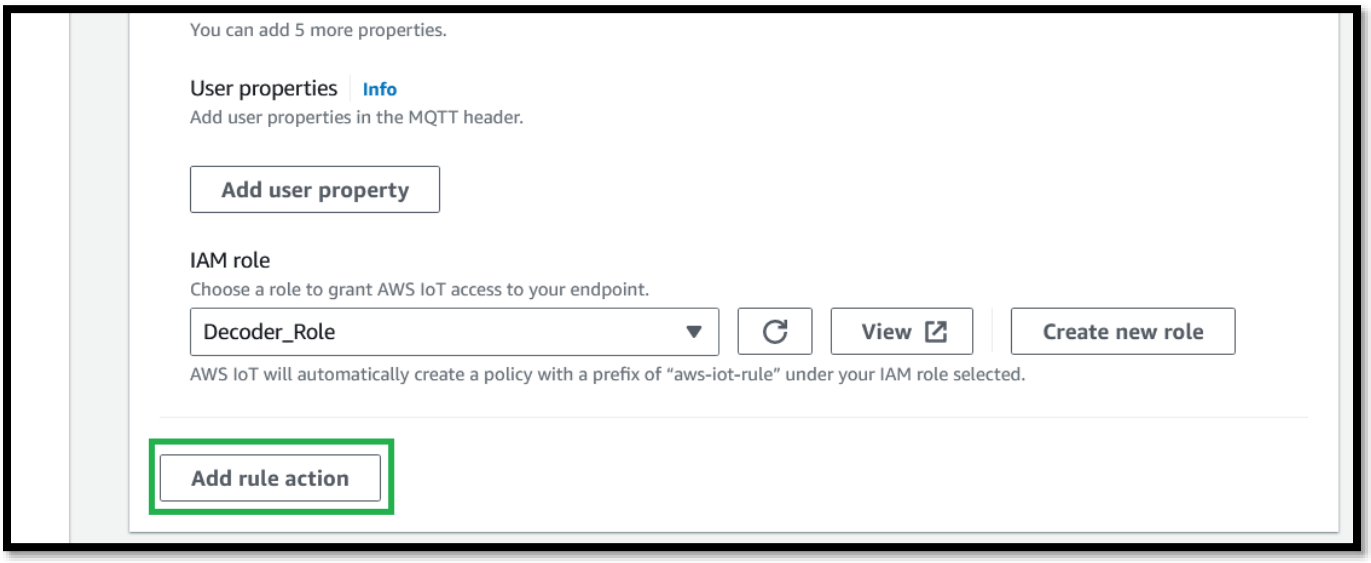

Figure 57: Adding the Lambda routing Action

In the Action dropdown, Lambda should be selected. The name of the Decoder Lambda should be selected, then Add rule action clicked, as shown in Figure 58.

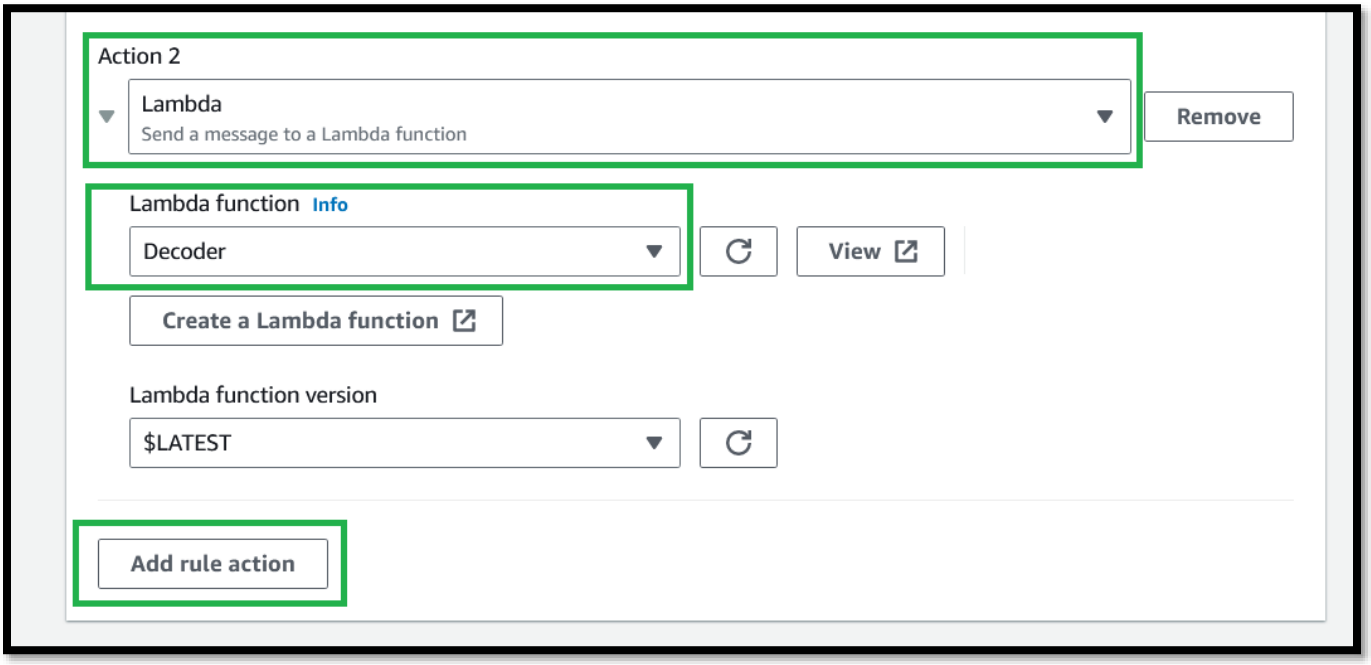

Figure 58: Setting Lambda rule action details

Click Next as shown in Figure 59 to proceed to the Review and create step.

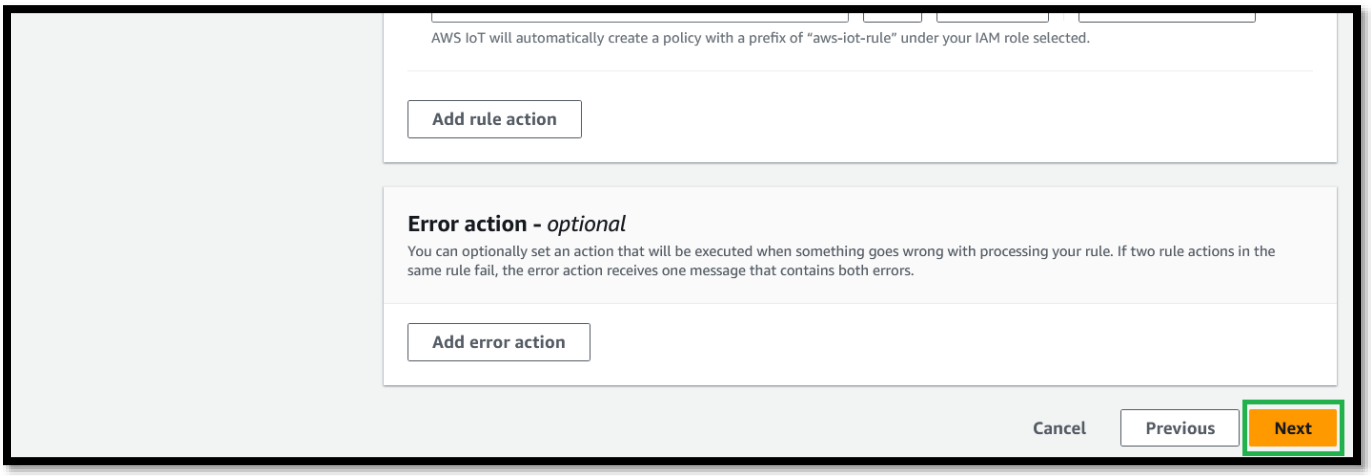

Figure 59: Finalising Attach rule actions

Click **Create** as shown in Figure 60 to complete adding the Decoder Rule.

Figure 60: Creating the Decoder Rule

9.1.3 Granting Lambda code access to the IoT Core

In the **Rules** menu, click the checkbox to the left of the Decoder rule and click **Activate**.

Note: In the previous step, adding the Action 'Send a message to a Lambda Function' updated the Policy for the Decoder Lambda to allow publishing of data to the IoT Core. Now the Policy has been updated, the Action can be removed from the Decoder Rule. This will not affect the updated Policy document.

9.1.4 Creating the Extractor Rule and Extracted Database

You must create a further Rule called "Extractor" as described in section 9.1.2. Set the query for this Rule as follows:

```
SELECT output.timestamp, output.devEUI, output.temperature FROM 'Decoded'
```

This Rule extracts the timestamp, Dev EUI and temperature from messages published to the Decoded topic.

Under the **Rule actions**, create a **Republish to AWS IoT topic** action, to the "Extracted" topic. This allows input to the database to be observed.

Add another action to the Rule to publish incoming data to the database. Select **DynamoDB**, then click **Create Dynamo DB table** as shown in Figure 61.

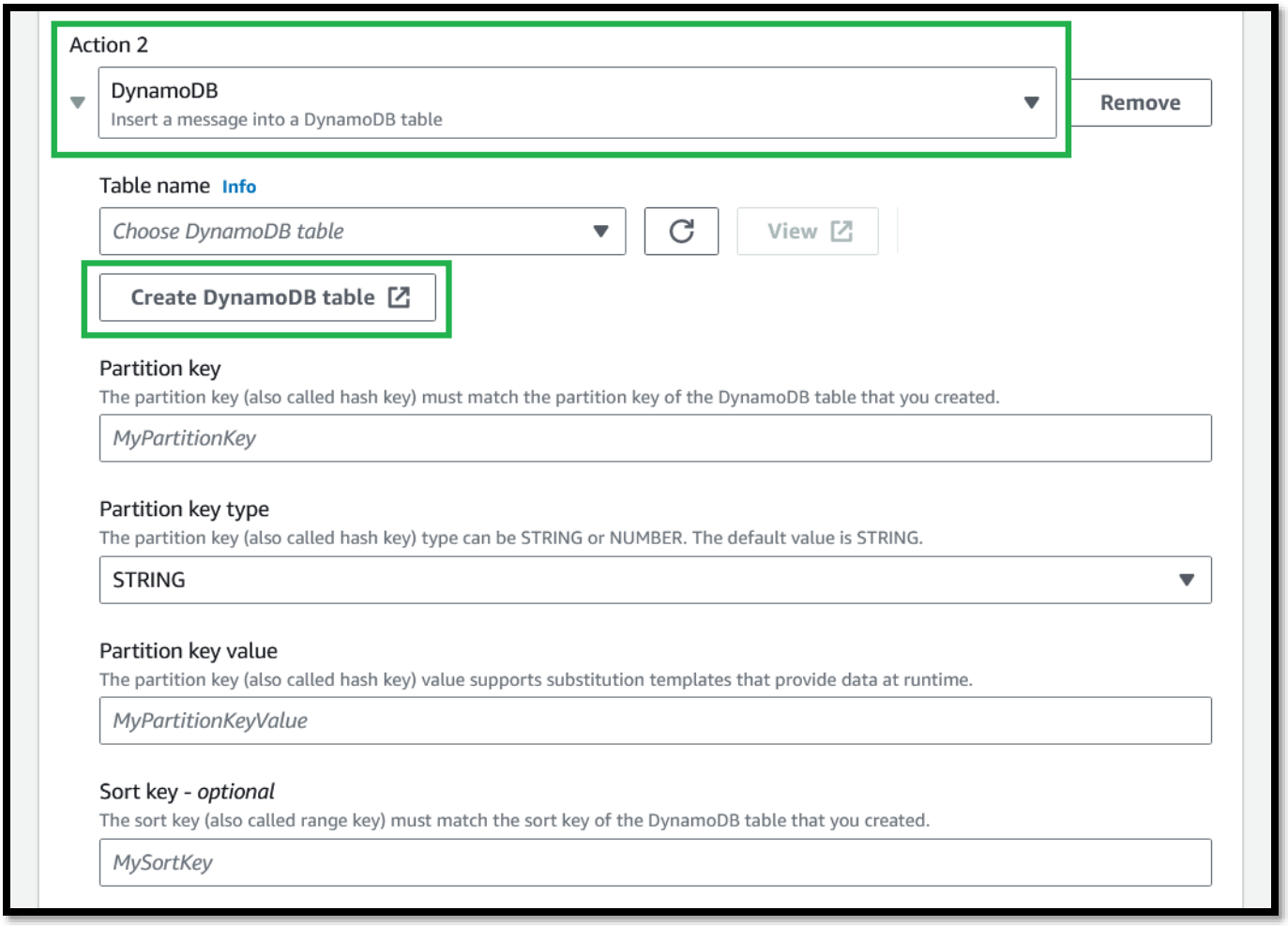

Figure 61: Adding the insert message into a DynamoDB table action

As shown in Figure 62, **Table name** should be set to **Extracted** and the **Partition Key** set to **Timestamp**.

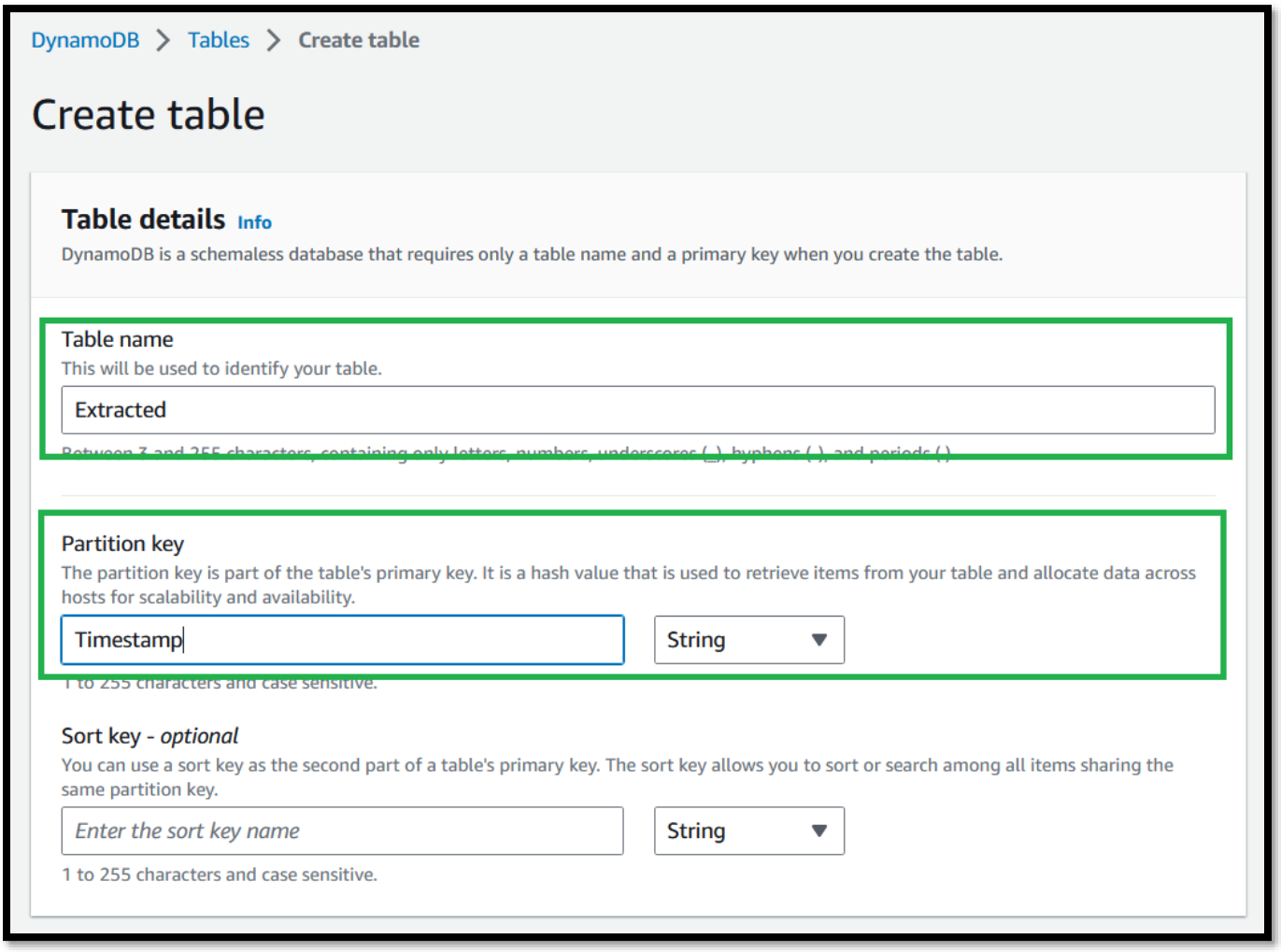

Figure 62: Creating the DynamoDB Extracted table

Create Table at the foot of the page can then be clicked, resulting in the dialog shown in **Figure 63**.

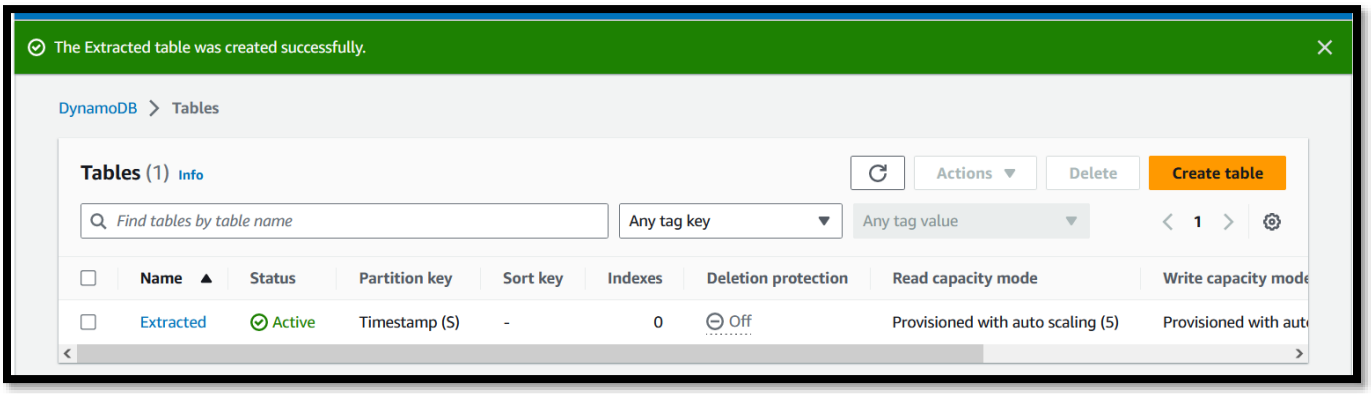

Figure 63: DynamoDB table summary

Returning to the **Extractor Rule**, the **Table name, Partition key** and **Partition key value** can now be entered as shown in **Figure 64**.

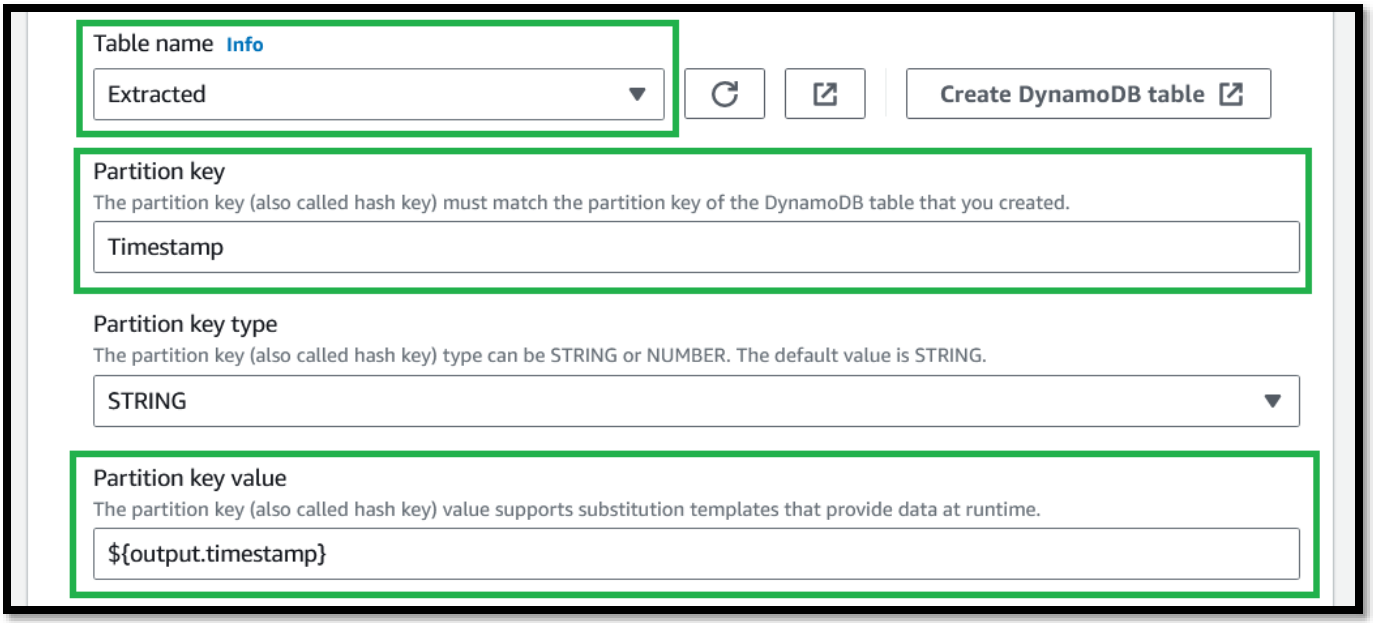

Figure 64: Setting DynamoDB table details

A **Role** must also be created, at the foot of the page, **Create new role** should be clicked, and **ExtractedDB_Role** entered as the Role name. **Create** can then be clicked, as shown in **Figure 65**.

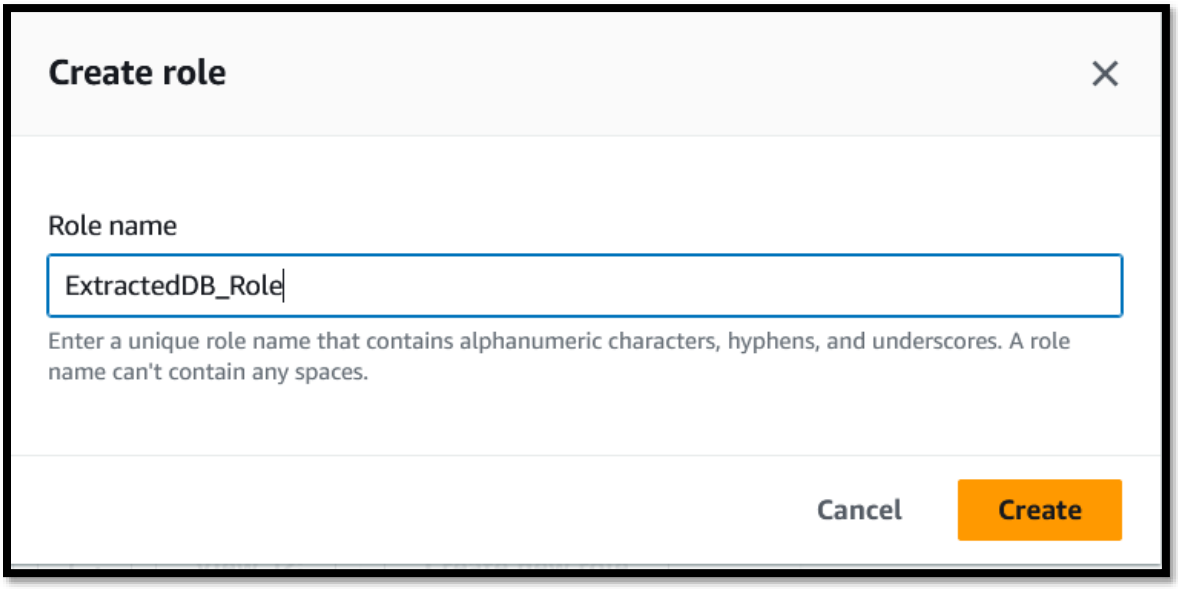

Figure 65: Creating the ExtractedDB role

At the foot of the page, **Add rule action** can now be clicked to finalise adding the Rule action, followed by **Update** to update the Rule.

9.1.5 Enabling the Decoder and Extractor Rules

Before you may invoke a Rule, it must be enabled within the Rules Engine. From the Rules main page, locate the Decoder and Extractor Rules as shown in Figure 66.

Figure 66: Enabling the Decoder and Extractor Rules

Click the checkbox to the left of each Rule and click **Activate**.

9.1.6 Testing the application

From the AWS IoT page, under **Test** in the left menu, click **MQTT test client** as shown in Figure 67.

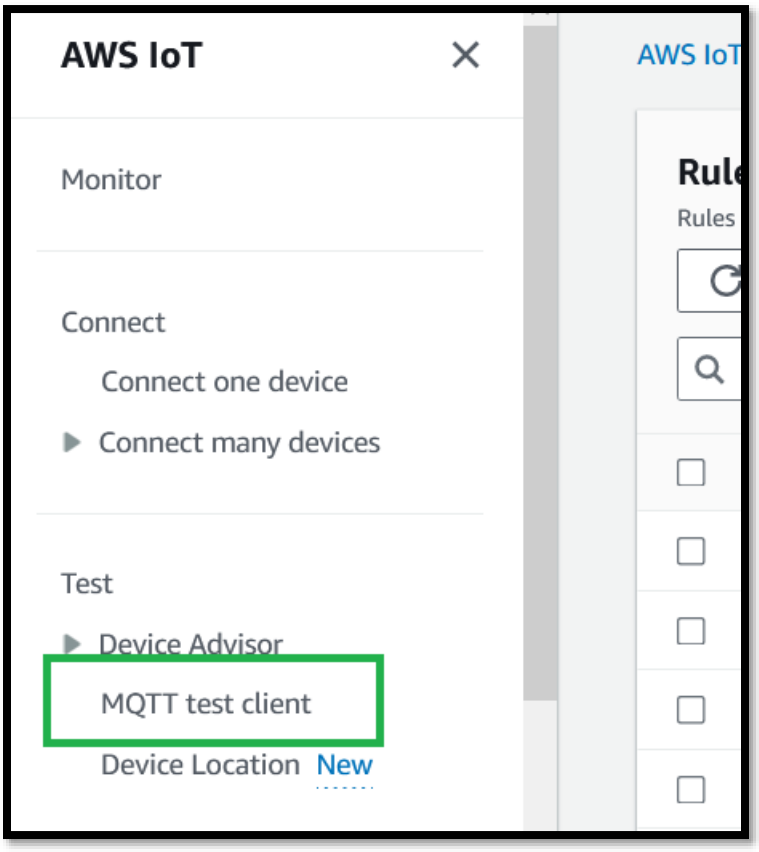

Figure 67: Opening the IoT Core MQTT Client

Click **Subscribe to a topic**, then set *Subscription Topic* to "Decoded" and click **Subscribe to topic** as shown in Figure 68.

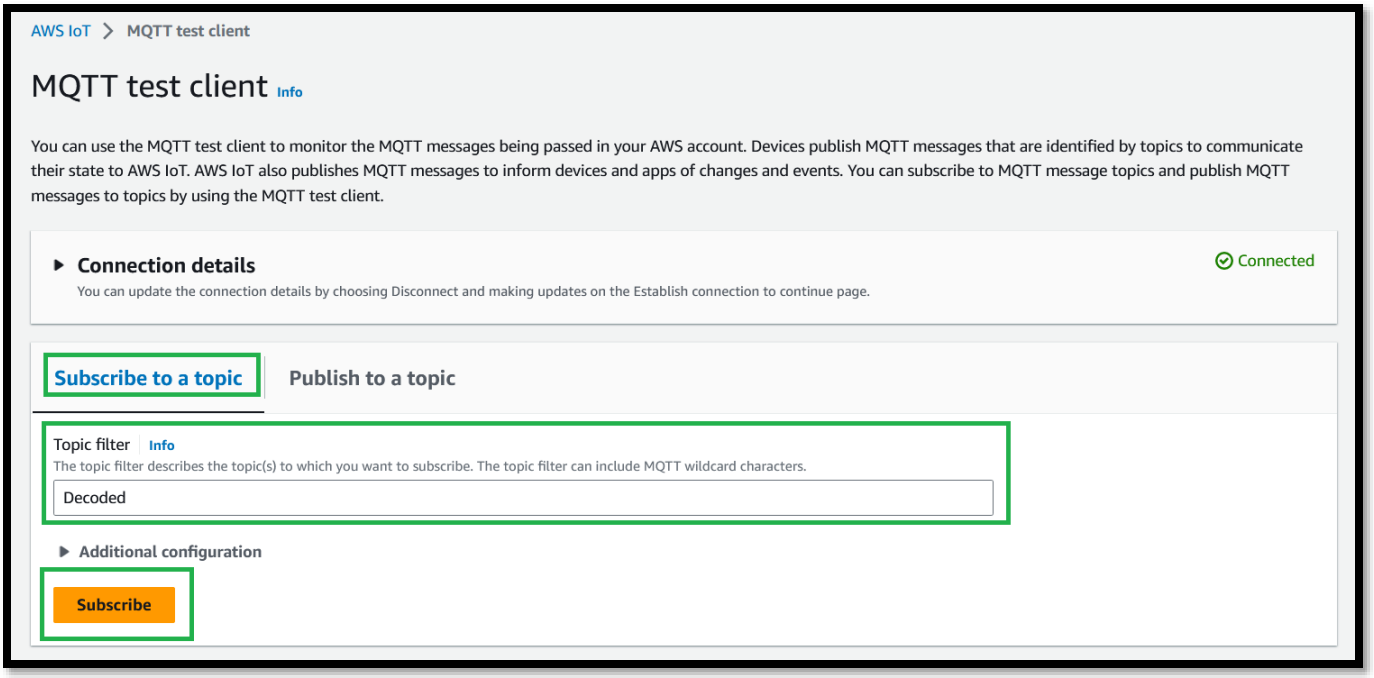

Figure 68: Subscribing to the Decoded topic

Repeat the process for the "Extracted" topic.

Available subscriptions are displayed at the foot of the MQTT Client page as shown in Figure 69.

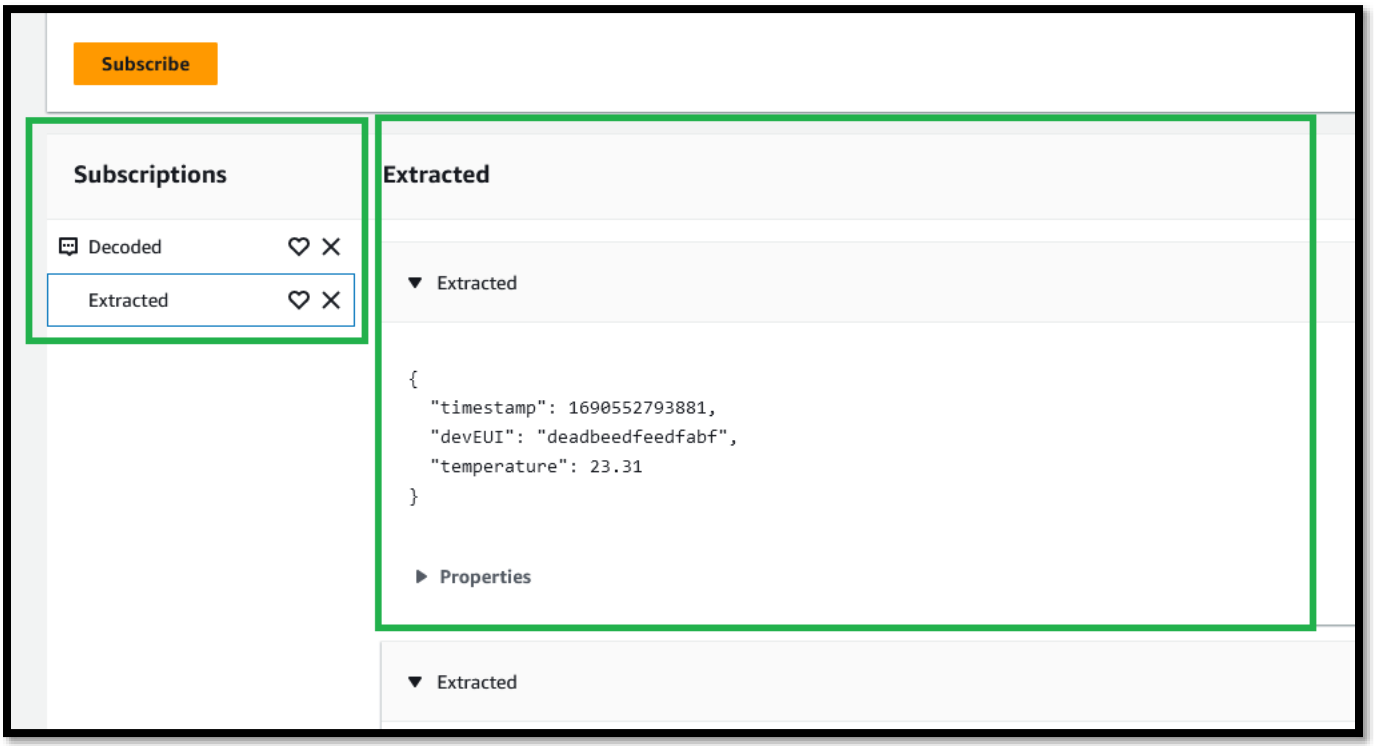

Figure 69: Available topic subscriptions

The DynamoDB service page allows you to inspect the content of the "Extracted" database. From the AWS Management Console, click **DynamoDB** as shown in Figure 70.

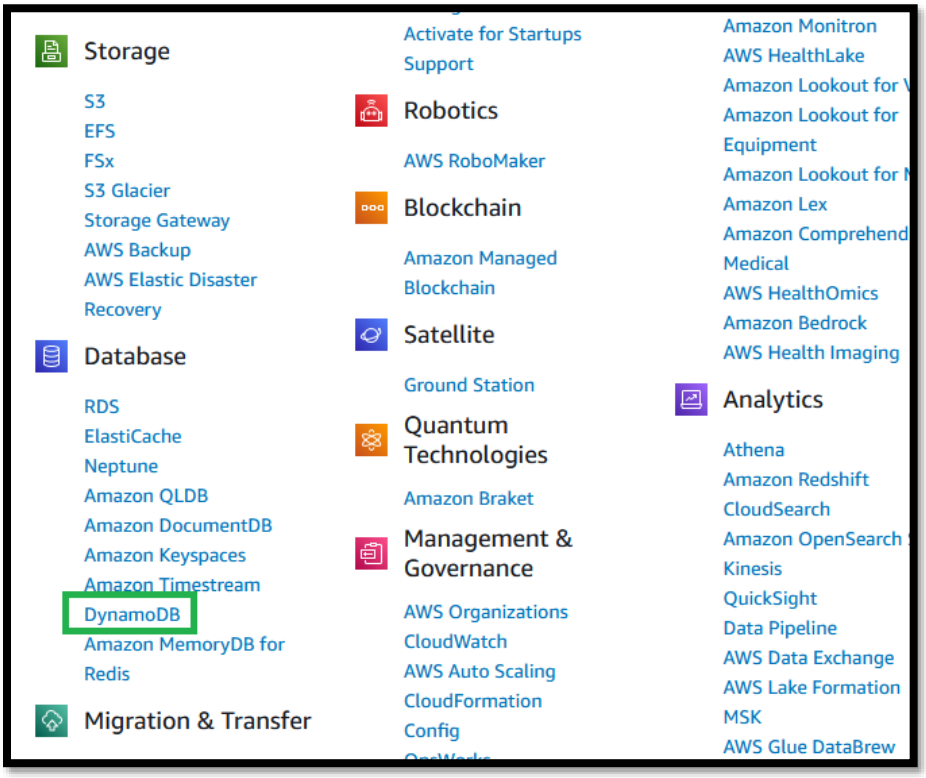

Figure 70: Opening the DynamoDB service from the AWS Management Console

Click **Tables** in the submenu on the left, then select *Extracted* as shown in Figure 71.

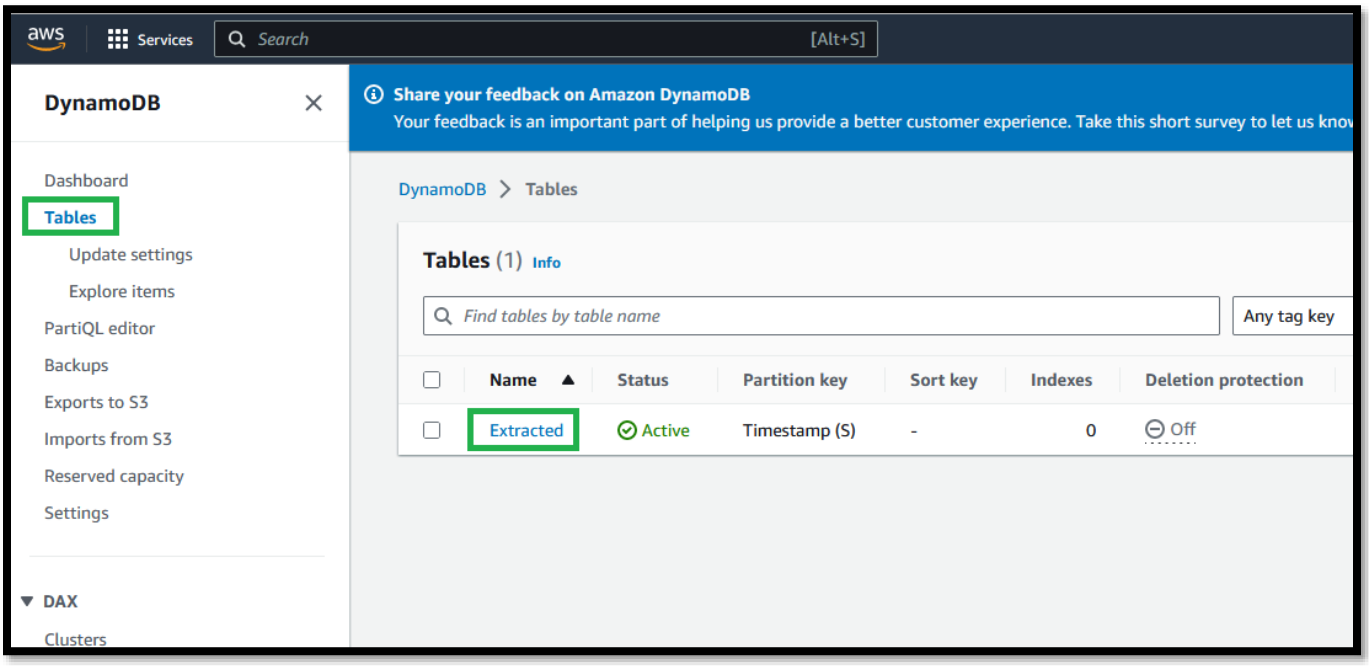

Figure 71: Opening the 'Extracted' database table

From the AWS MQTT Client, inspect the "Extracted" topic messages to review the data being transferred to the database as shown in Figure 72.

Figure 72: Extracted topic messages

From the DynamoDB page, click **Items** to view the **Extracted** table, then **Explore table items** to view the table contents as shown in Figure 73.

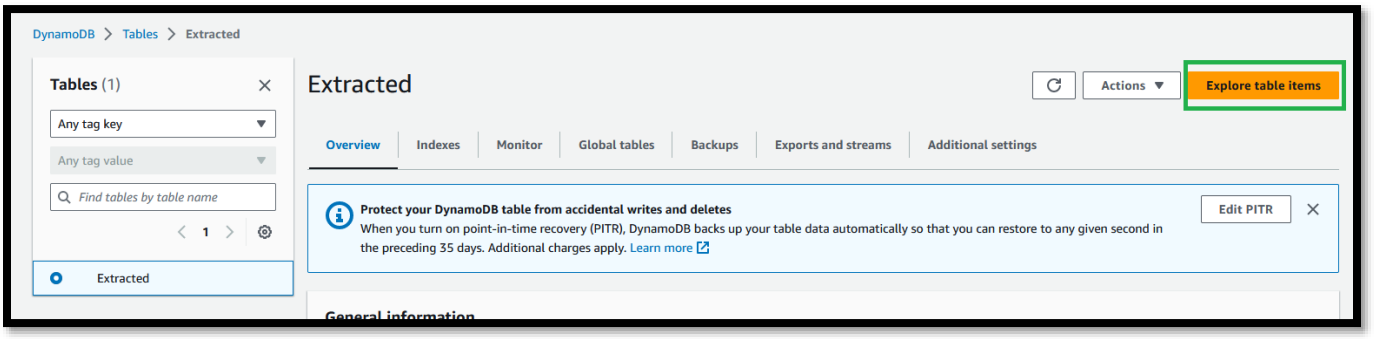

Figure 73: Viewing Extracted topic messages being added to the Extracted table

There should be parity between the content of the AWS MQTT Client and the table, as shown in Figure 74.

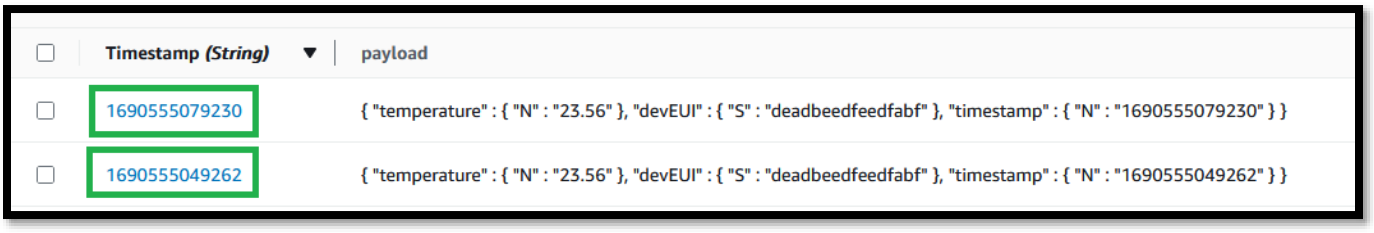

Figure 74: Extracted table contents

9.2 Cayenne Protocol Format example

Before implementing this example, set the sensor Packet Format to Cayenne. Further details of the Cayenne Low Power Protocol are available in the Cayenne Low Power Protocol description at reference [V]. Lambda code in NodeJS v16.x format for decoding the data packets is available from our GitHub page at reference [W].

Figure 75 shows the application architecture. Messages from the sensor are passed to the Decoder Rule, which invokes the Decoder Lambda function. This extracts the message payload data and decodes it into meaningful values. These are then published to the Decoded Topic. The published messages can be inspected by subscribing to the topic using AWS' MQTT Client.

Note this application uses the Cayenne Packet Format for the purposes of demonstration only and is only bound to the Cayenne data format via the Lambda code. Substituting the Laird Decoder code will result in the same functionality for sensors with a Packet Format configuration of 'Laird 1' or 'Laird 2'.

A second rule, Warning Rule, subscribes to the Decoded Topic, and publishes messages to a second topic, Warning Topic, when any temperature values are found to be less than ten degrees. When published, an email is set via an SNS connection to warn of the temperature falling below this value.

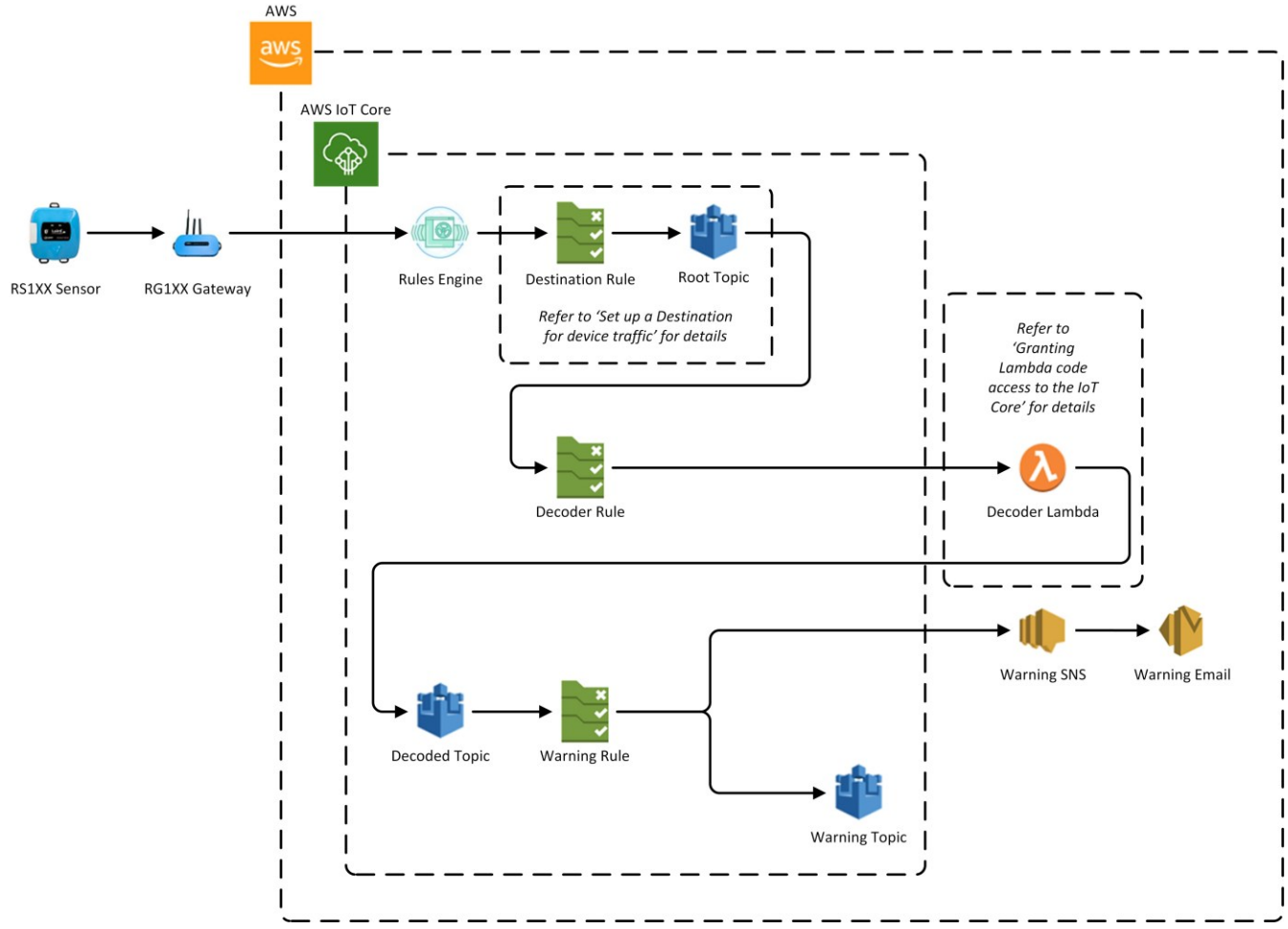

Figure 75: Cayenne Packet Format application architecture

9.2.1 Creating the Decoder Lambda function

Follow the steps in section 9.1.1 using the library_cayenne.js, index.js and sensor_types_cayenne.js files in place of library_laird.js, index.js and messages_laird.js from the Laird Decoder folder.

9.2.2 Creating the Decoder Rule and Decoded topic

Repeat the steps in section 9.1.2 and 9.1.3.

9.2.3 Creating the Warning Notification

The Simple Notification Service (SNS) sends notifications when triggered, in this case emails to a subscribed address when the sensor temperature falls below a certain value.

To create a Simple Notification, from the AWS main page, click **Simple Notification Service** as shown in Figure 76.

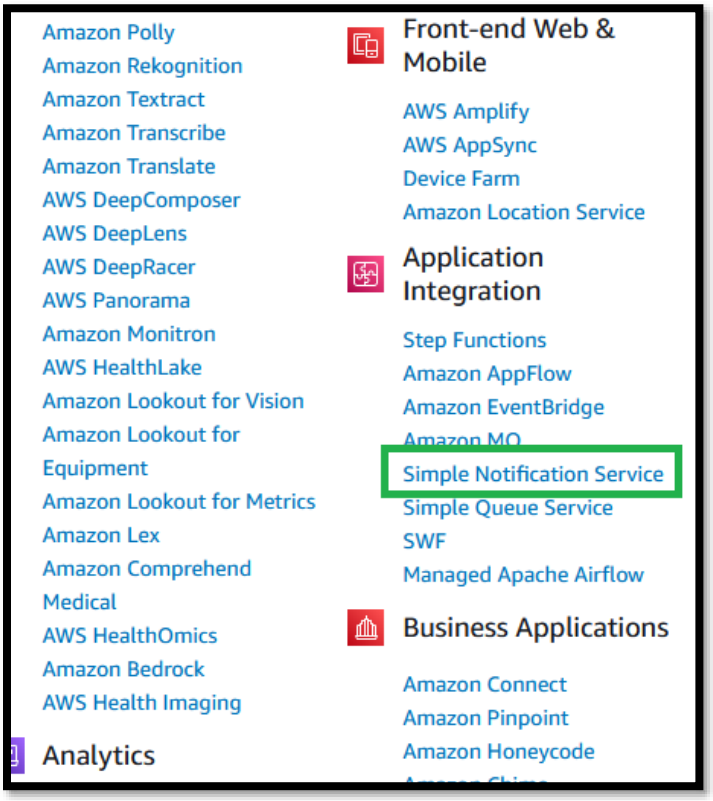

Figure 76: Opening the SNS page

This opens the Simple Notification Service page as shown in Figure 77. In the left submenu click **Topics**.

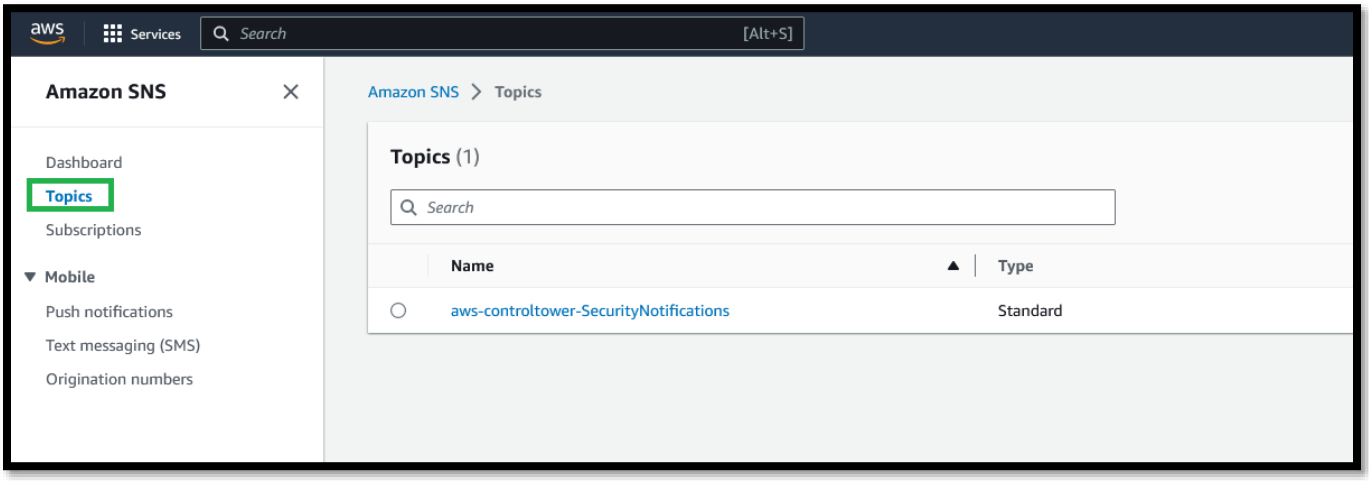

Figure 77: SNS main page

Click **Create topic** as shown in Figure 78. This facilitates sending emails when appropriate messages are published.

Figure 78: SNS Topic list

In the Create topic window, set *Type* to standard and *Name* to "Warning". Click **Create topic**.

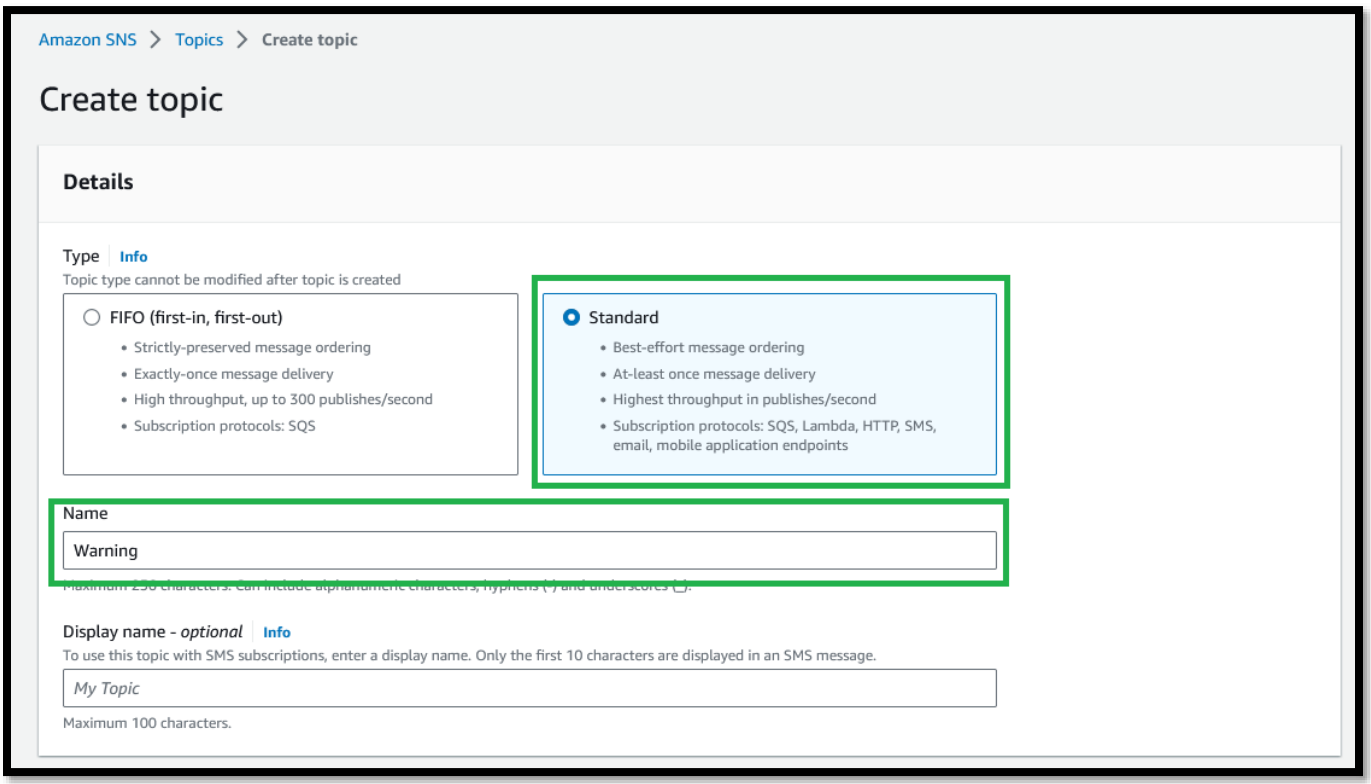

Figure 79: Creating an SNS topic

If successful, the new topic page displays as shown in Figure 80. Note the ARN of the topic for later use.

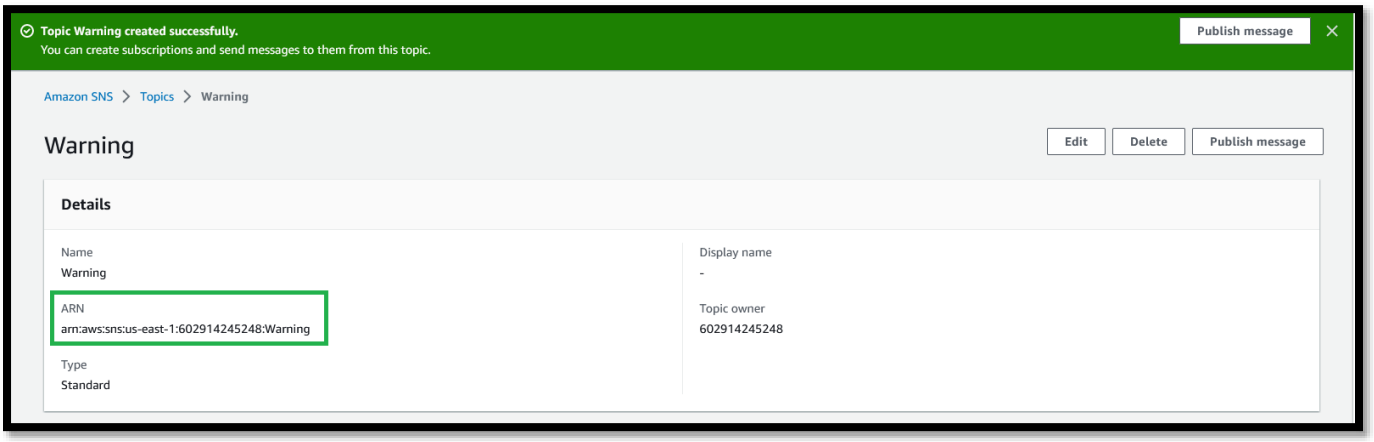

Figure 80: Successful creation of the Warning topic

Scroll down to the details of the Subscriptions to the topic as shown in Figure 81. Click **Create subscription** to add details of the warning email recipient.

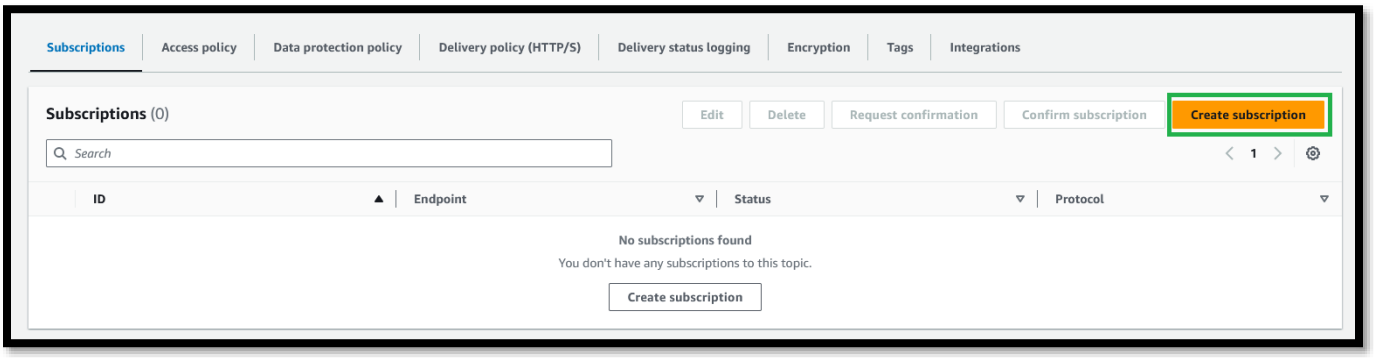

Figure 81: Warning topic subscriptions

Enter details as shown in Figure 82. Set *ARN* to the ARN of the Warning topic noted previously. Set *Protocol* to "Email" and *Endpoint* to the intended recipient email address. Click **Create subscription** to create the subscription.

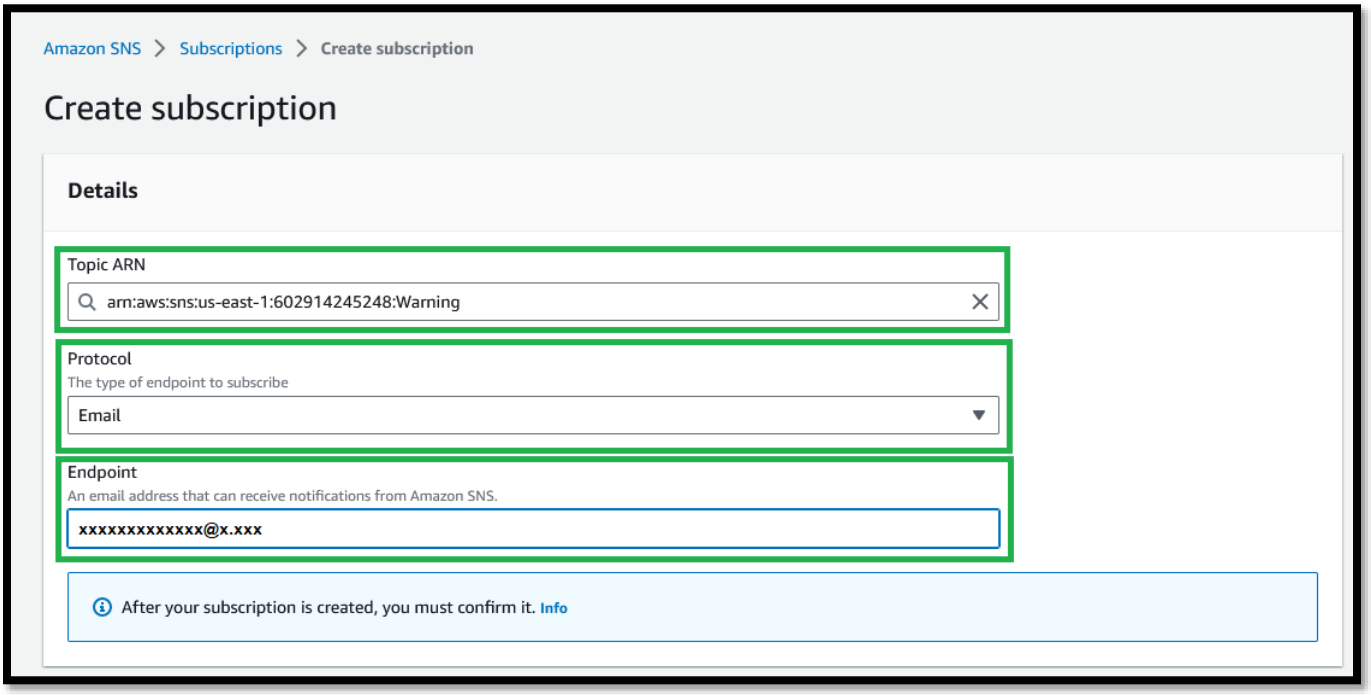

Figure 82: Warning topic subscription details

Before the service can send notifications, the recipient email subscription must be confirmed. To confirm the subscription, click the verification link in the verification email, which is sent to the email address when the subscription is created. The status is displayed as pending, as shown in Figure 83, until the subscription is verified.

Figure 83: Pending confirmation of the Warning topic subscription

The verification email is as shown in Figure 84.

Figure 84: SNS subscription confirmation email

Having confirmed the subscription, the details appear as shown in Figure 85.

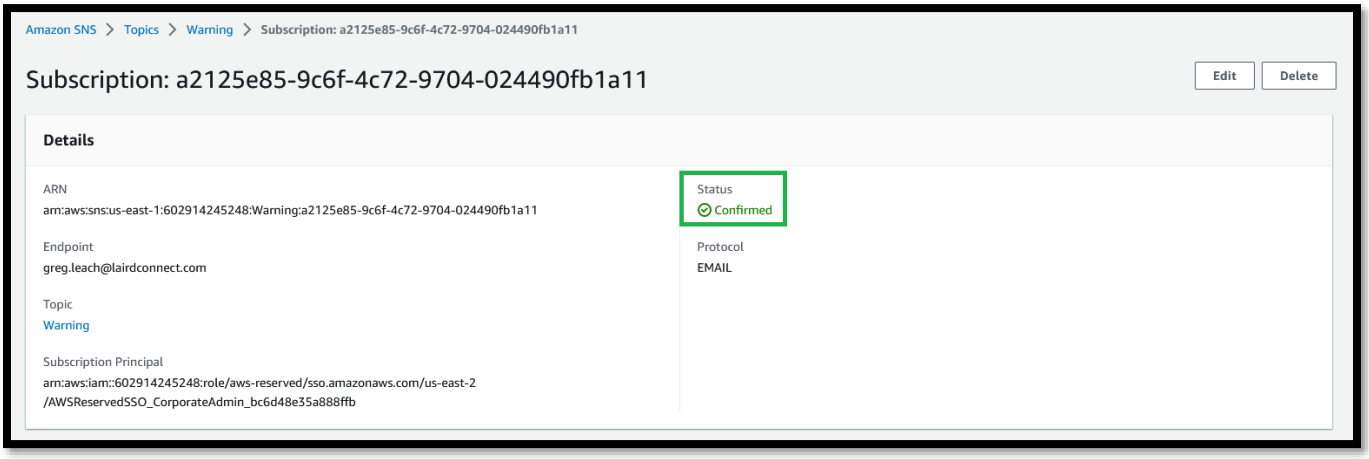

Figure 85: Confirmation of the Warning topic subscription

9.2.4 Creating the Warning Rule and topic and SNS Action

At the **IoT Core** page, from **Message routing**, create a **Rule** called **Warning**. This will only publish messages when the sensor temperature falls below 40˚C. Create the rule as described in section 9.1.4. The query is as follows:

```
SELECT * FROM 'Decoded' WHERE output.temperature < 40
```

This query ensures messages are only published when the temperature falls below 40˚C.

Add an action to the republish the data to the 'Warning' topic. This allows data being sent to the email recipient to be observed.

Add a **Rule action Send a message as an SNS push notification** as shown in Figure 86.

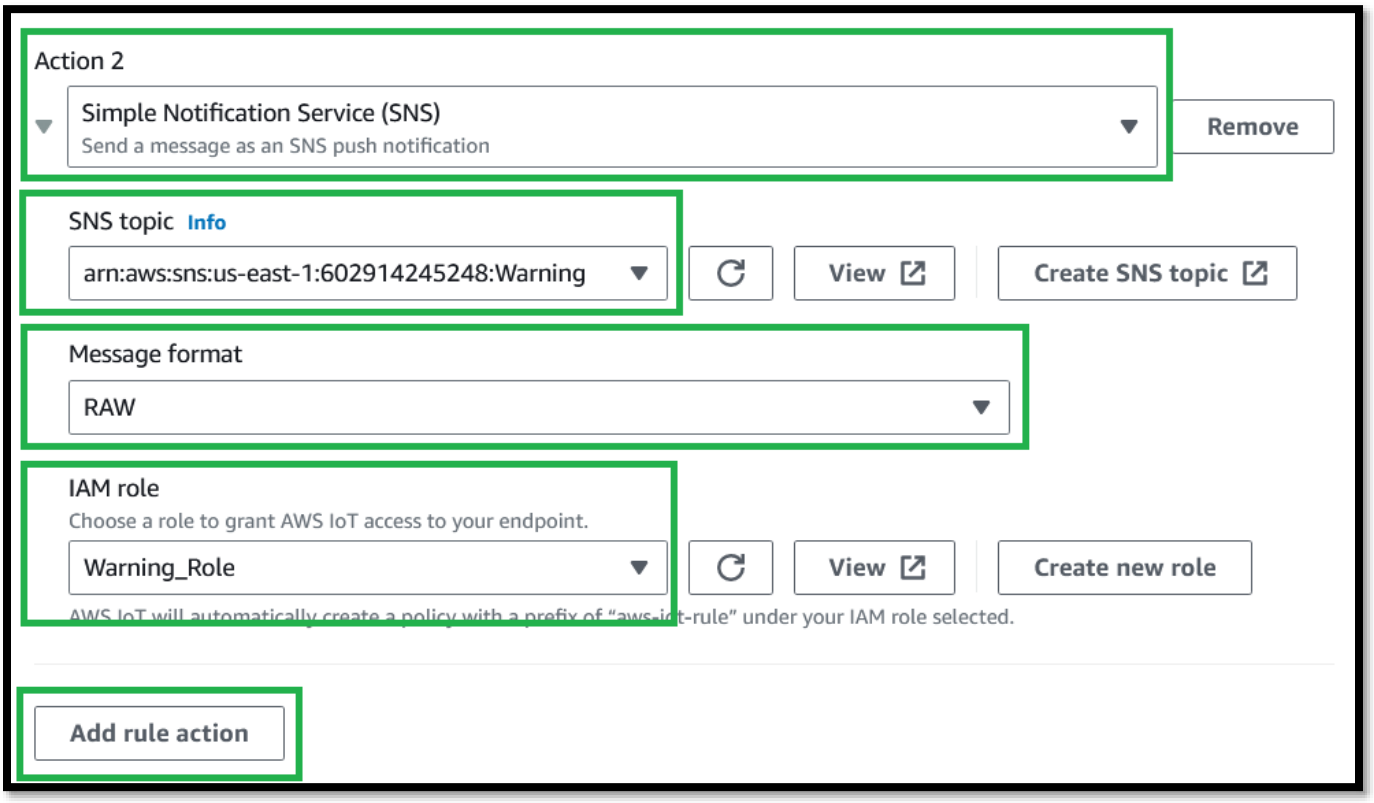

Figure 86: Adding the SNS push notification

Click **Add rule action** to finalize adding the Rule action. Then enable the Rule.

9.2.5 Testing the application

Use the AWS MQTT Client to subscribe to the "Decoded" and "Warning" topics. Messages are only published to the "Warning" topic in event of the temperature falling below 40°C. For each message published to the 'Warning' topic as it's shown in Figure 87, an email should be received as shown in Figure 88.

Figure 87: Message received by 'Warning' topic

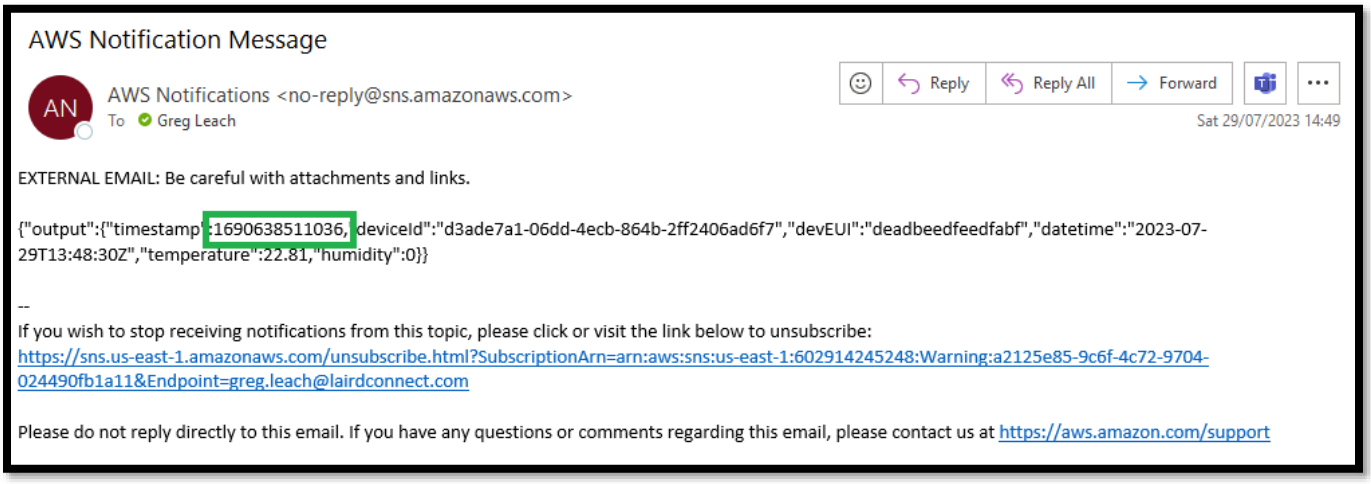

Figure 88: Email message received due to publish to 'Warning' topic

10 Gateway OTA Updates

The RG1xx gateways support over-the-air firmware updates. The following steps describe how to update the gateway firmware.

10.1 Starting the firmware update

From the web interface Dashboard, click **Settings** as shown in Figure 89.

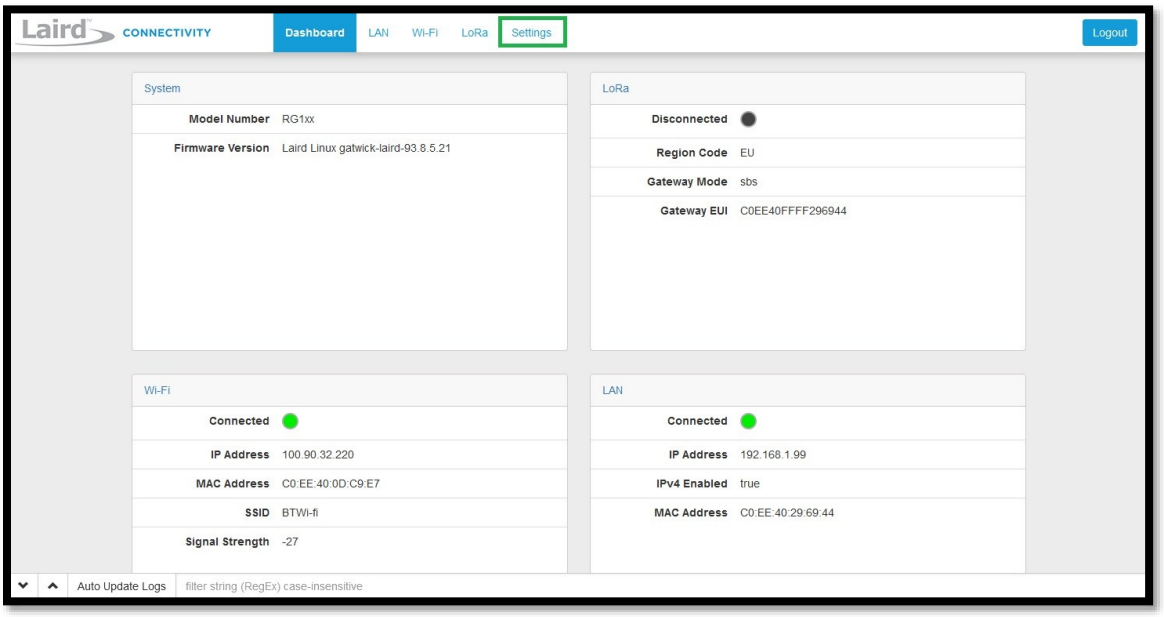

Figure 89: Opening the gateway Settings page

From the Settings page, click **Update Firmware** as shown in Figure 90.

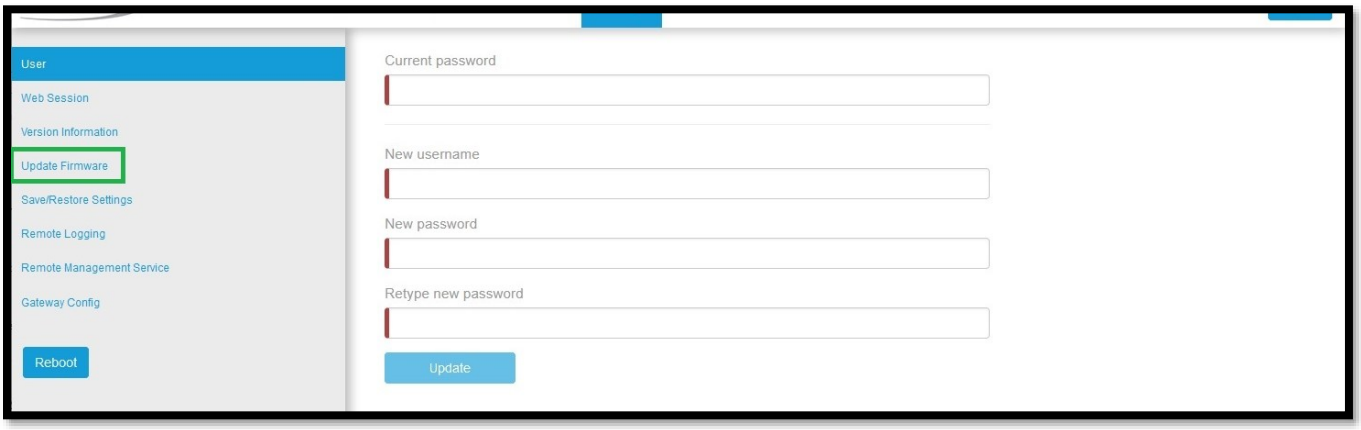

Figure 90: Opening the Update Firmware page

The Update Firmware page is shown in Figure 91.

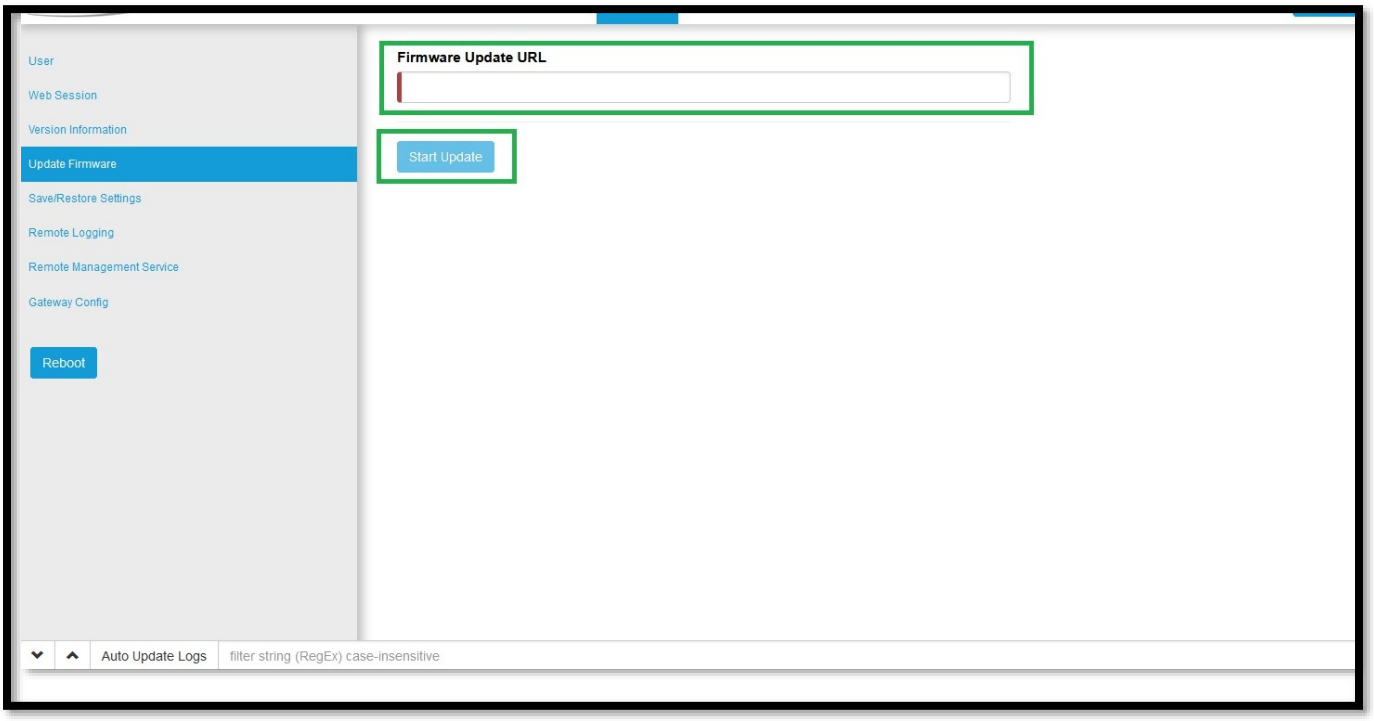

Figure 91: Update Firmware page

Enter the URL for the firmware version required.

Note: Depending upon the firmware currently in use, updates may first be required to a previous release.

After entering the firmware URL, click **Start Update** to begin the firmware update.

10.2 Firmware URLs

The following are correct as of December 2020. Refer to the appropriate gateway User Guide [A], [B] such that a newer firmware version may be available.

Note: The following list the upgrade URLs based on what firmware is currently running on the gateway. This is an important step, as some firmware versions require updating to an intermediate firmware before updating to the final firmware. **Carefully follow the steps based on the firmware that is currently running on your gateway**.

10.2.1 Firmware Version 93.7.1.13 (GA1)

If the gateway is running version 93.7.1.13, the user should use the following link to upgrade to the next version.

https://www.lairdtech.com/products/rg1xx-lora-gateway/firmware/GA1.1/fw.txt

After updating with this link, the gateway will be running version 93.7.1.14. The instructions for that version should then be followed to update to the latest version of firmware.

10.2.2 Firmware Version 93.7.1.14

If the gateway is running version 93.7.1.14, the user should use the following link to upgrade to the next version.

https://www.lairdtech.com/products/rg1xx-lora-gateway/firmware/GA2.1/fw.txt

After updating with this link, the gateway will be running version 93.7.2.10. The instructions for that version should be used to update to the latest version of firmware.

Note that this upgrade performs a factory reset on the gateway, necessitating repeating the gateway setup.

10.2.3 Firmware Version 93.7.2.9 (GA2)

If the gateway is running version 93.7.2.9, the user should use the following link to upgrade to the next version.

https://www.lairdtech.com/products/rg1xx-lora-gateway/firmware/GA2.1/fw.txt

After updating with this link, the gateway will be running version 93.7.2.10. The instructions for that version should be used to update to the latest version of firmware.

Note that this upgrade performs a factory reset on the gateway, necessitating repeating the gateway setup.

10.2.4 Firmware Version 93.7.2.10 (GA2.1)

If the gateway is running version 93.7.2.10, the user should use the following link to upgrade to the next version.

https://www.lairdtech.com/products/rg1xx-lora-gateway/firmware/newest/fw.txt

Note this requires users to manually update the URL. After updating with this link, the gateway will be running GA3 firmware (93.7.3.x) or newer. The instructions for that version should be followed to update to the latest version of firmware.

10.2.5 Firmware Version 93.7.3.4 (GA3 and newer)

GA3 firmware (93.7.3.x) and newer versions have a feature to automatically notify the user if new firmware is available and where to download the firmware.

10.2.6 Firmware Version 93.8.4.28 (GA4) & 93.8.4.37 (GA4.1)

The user should use the following link to upgrade to the next version.

https://www.lairdtech.com/products/rg1xx-lora-gateway/firmware/GA4.1/fw.txt

10.2.7 Firmware Version 93.8.5.18 (GA5) & 93.8.5.21 (GA5.1)

This is the latest production release.

10.2.8 Firmware Version 93.8.5.25 (GA5.2)

The user should use the following link to upgrade to the next version. This is the minimum required release.

https://connectivity-firmware.s3.amazonaws.com/rg1xx-lora-gateway/firmware/93.8.5.25/fw.txt

10.3 Firmware update process

Click **Start** Update. Details of the update process appear on the Update Firmware page as shown in Figure 92.

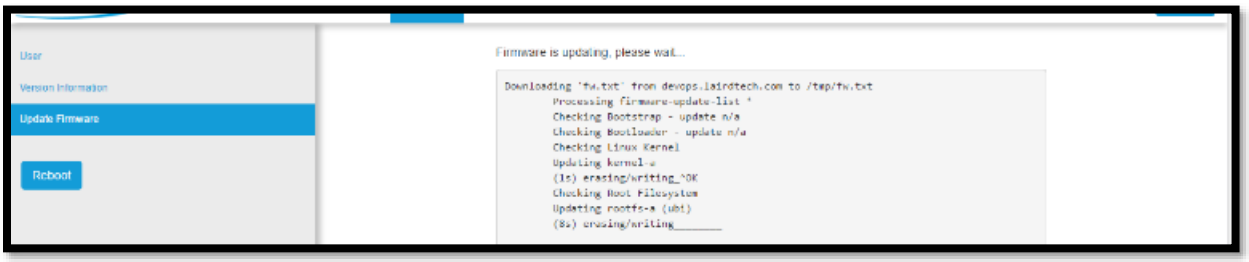

Figure 92: Firmware update progress display

Upon completion of the update, the page prompts you to reboot the gateway as shown in Figure 93.Click **Reboot**. Upon restart, if there are more steps in the firmware upgrade for your software version, repeat the process until you've updated to the desired firmware.

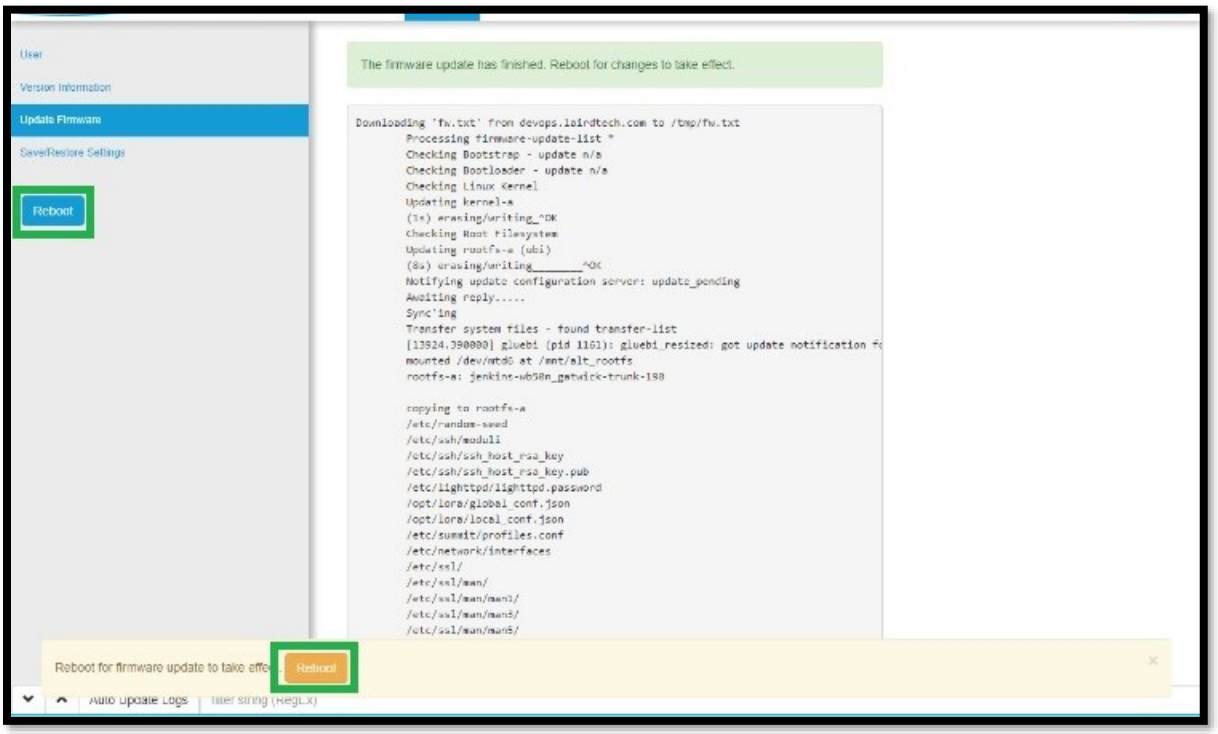

Figure 93: Reboot prompt following firmware update

11 Sensor OTA Updates

Sensor firmware is updated using the Sentrius sensor mobile app. From the mobile app main page, tap **Device FW Update** as shown in Figure 94.

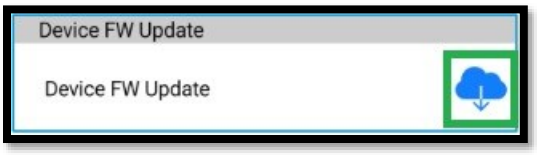

Figure 94: Device FW Update icon

The app displays a list of available firmware versions as shown in Figure 95. Ezurio recommends you upgrade to the latest version available. Tap a firmware version number to begin the update. This process typically takes less than one minute.

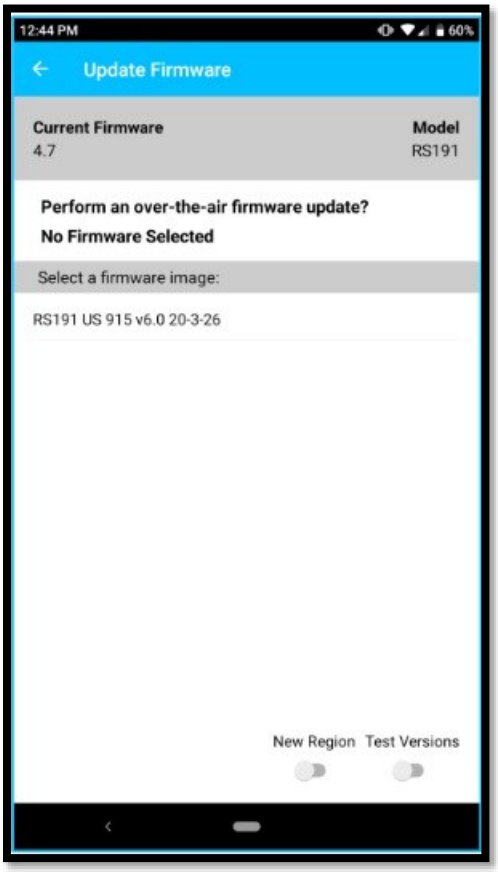

Figure 95: Device FW Update available images

12 Debugging

The following describe debugging methods available for integrating the gateway.

The first is an activity log. From the gateway web interface, click the up arrow in the lower right-hand corner as shown in Figure 96 to partially reveal the log window. Clicking again will further reveal the window.

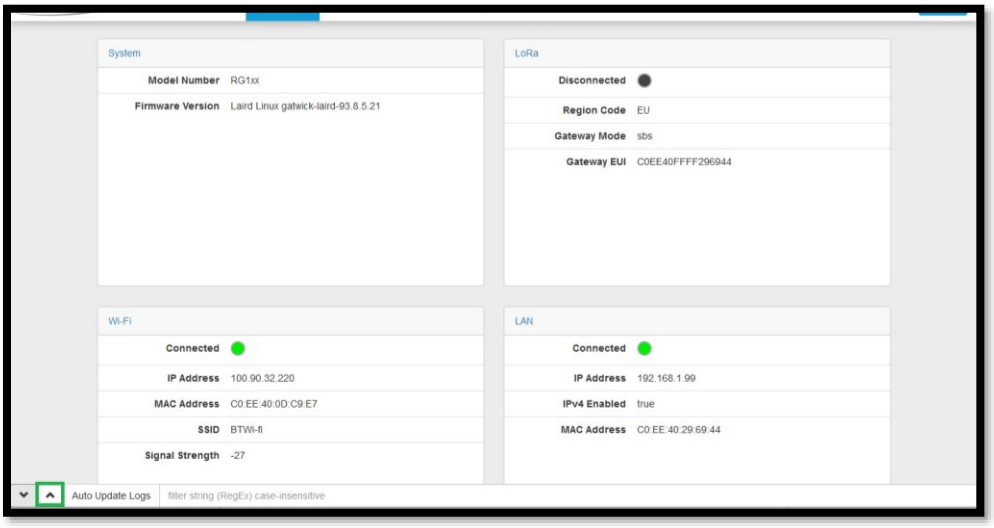

Figure 96: Enabling the gateway log

Click **Auto Update Logs** to auto-refresh the log as shown in Figure 97.

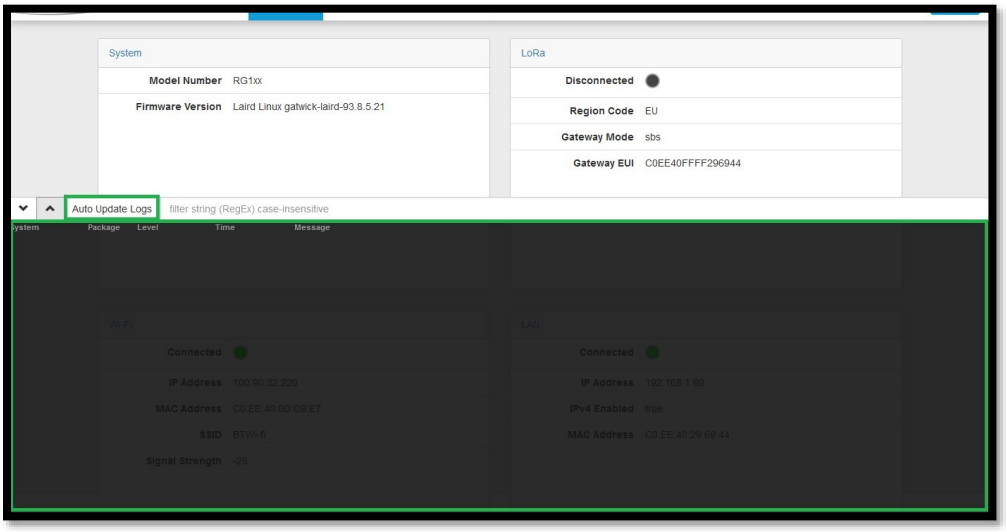

Figure 97: Gateway log window

The log window is continuously updated with details of activities being performed by the gateway as shown in Figure 98.

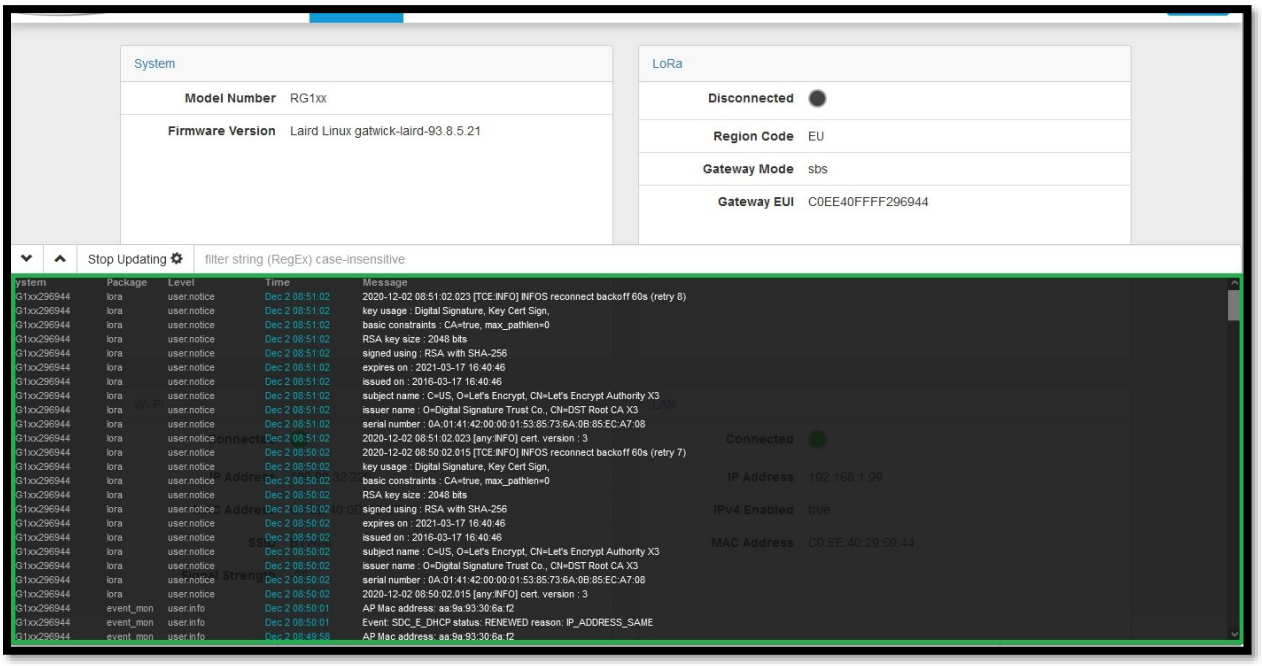

Figure 98: Update of the gateway log window

13 Troubleshooting

When the gateway is successfully connected to the AWS IoT Core for LoRaWAN LNS, the LoRa section of the dashboard displays as connected. When not connected, it displays as disconnected, as shown in Figure 99.

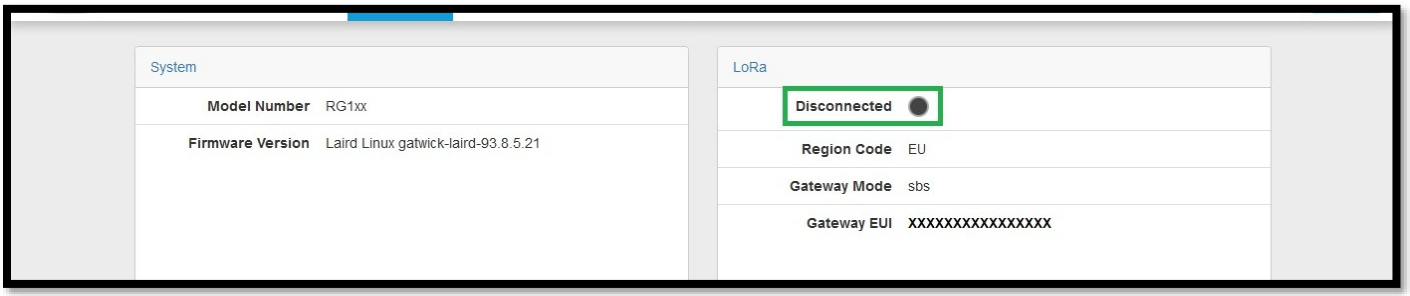

Figure 99: Gateway LoRa connection indication

If the connection fails, check the following.

- Verify that certificate details are correct
- Verify that endpoint details point at the correct region and port in BasicsStation setup
- Verify that gateway region matches the region managed by the AWS IoT Core for LoRaWAN endpoint
- Verify that the correct EUI is used to generate the ARN for the gateway

14 RS1xx Device Profile Specification

Table 3 summarises parameters required when creating Device profiles for the RS1xx range of sensors.

RS1xx region	Device profile settings						
	Frequency Band	MAC version	Regional parameters version	MaxEIRP	Supports Class B	Supports Class C	Supports Join
Australia (AU915)	AU915	1.0.2	RP002-1.0.1	16	Disabled	Disabled	Enabled
Australia (AS923)	AS923-1						
Europe	EU868						
Hong Kong	AS923-1						
New Zealand	AS923-1						
North America	US915						
Singapore	AS923-1						
Taiwan	AS923-1						

Table 3: RS1xx Device profile settings

15 References

Ref	Description
[A]	User Guide – RG1XX https://www.lairdconnect.com/documentation/user-guide-RG1xx
[B]	User Guide – RG1XX & LTE https://www.lairdconnect.com/documentation/user-guidedatasheet-rg191lte
[C]	User Guide – AWS IoT Core for LoRaWAN https://console.aws.amazon.com/iotwireless/home?region=us-east-1
[D]	How to use LoRa Basics Station https://lora-developers.semtech.com/library/tech-papers-and-guides/how-to-use-lora-basics-station/
[E]	Product Brief – Sentrius RG1XX Series Gateway https://www.lairdconnect.com/documentation/product-brief-sentrius-RG1xx-series-gateway
[F]	Product Brief – Sentrius RG191 + LTE https://www.lairdconnect.com/documentation/product-brief-sentrius-rg191lte
[G]	Product Brief – Sentrius RS1XX External RTD Temperature Probe https://www.lairdconnect.com/documentation/product-brief-sentrius-rs1xx-external-rtd-temp-probe
[H]	Product Brief – Sentrius RS1XX External Temperature Sensor https://www.lairdconnect.com/node/11142
[I]	Product Brief – Sentrius RS1XX Integrated Temperature & Humidity Sensor https://www.lairdconnect.com/documentation/product-brief-sentrius-rs1xx-integrated-temp-humidity-sensor
[J]	Product Brief – Sentrius RS1XX with Open/Closed Sensor & Integrated Temperature & Humidity Sensor https://www.lairdconnect.com/documentation/product-brief-sentrius-rs1xx-openclosed-sensor-and-integrated-temprh
[K]	RG1XX Certifications https://www.lairdconnect.com/wireless-modules/lorawan-solutions/sentrius-RG1xx-lorawan-gateway-wi-fi-ethernet-optional-lte-us-only
[L]	AWS Set up your AWS account https://docs.aws.amazon.com/iot/latest/developerguide/setting-up.html

Ref	Description
[M]	AWS Example IoT Policies https://docs.aws.amazon.com/iot/latest/developerguide/example-iot-policies.html
[N]	AWS Security Best Practices https://docs.aws.amazon.com/iot/latest/developerguide/security-best-practices.html
[O]	Configuration Guide – RS1XX https://www.lairdconnect.com/documentation/sentrius-rs1xx-configuration-guide-v112
[P]	User Guide – RS1XX Open/Closed & Internal Temperature/Humidity Sensor https://www.lairdconnect.com/documentation/sentrius-rs1xx-ext-openclosed-and-int-temphumidity-sensor-user-guide-v11
[Q]	User Guide – RS1XX External Temperature Sensor https://www.lairdconnect.com/node/11151
[R]	RS1XX Certifications https://www.lairdconnect.com/wireless-modules/lorawan-solutions/sentrius-rs1xx-lora-enabled-sensors
[S]	Join EUI Guidance https://lora-developers.semtech.com/library/tech-papers-and-guides/the-book/joineui
[T]	Security Keys Guidance https://lora-developers.semtech.com/library/tech-papers-and-guides/the-book/security-keys/
[U]	Protocol Description – RS1XX https://www.lairdconnect.com/documentation/application-note-rs1xx-lora-protocol
[V]	Cayenne Low Power Protocol https://developers.mydevices.com/cayenne/features/
[W]	Laird Github page for RS1XX Sentrius integration https://github.com/LairdCP/RS1XX_AWS
[X]	AWS LoRaWAN Developer Guide https://docs.aws.amazon.com/iot/latest/developerguide/connect-iot-lorawan.html
[Y]	
[Z]	
[A1]	
[B1]	
[C1]	

16 Additional Information

Please contact your local sales representative or our support team for further assistance:

Headquarters	Ezurio 50 S. Main St. Suite 1100 Akron, OH 44308 USA
Website	http://www.ezurio.com
Technical Support	http://www.ezurio.com/resources/support
Sales Contact	http://www.ezurio.com/contact

Note: Information contained in this document is subject to change.

Ezurio's products are subject to standard Terms & Conditions.