AWS IoT Core for LoRaWAN

Sentrius™ RG1xx

Version 1.2

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 Glossary entries. Corrected filename references in sections 6.1.1 and 6.2.1. Clarified section 4.4.3 for source of Destination Role. Added further detail to section 4.4.3 to describe creation of the root MQTT topic for other Actions and Services. Added further detail to section 6 to clarify the initial source of data for created Rules. Added section 6.1.3 to describe update of the Role used to grant Lambda code access to the IoT Core. Updated Figure 30 and Figure 69 to emphasise Lambda Engine boundary and the requirement for the access Policy to be updated via IAM.	Greg Leach	Chris Boorman

CONTENTS

1 Introduction

This application note describes the steps involved in integrating Laird Connectivity's RG1xx series of gateways with AWS IoT Core for LoRaWAN. It is intended to be referred to in conjunction with the RG1xx User Guide [A] & RG1x User Guide (LTE) [B], which further describe RG1xx 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 5.2, "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
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.

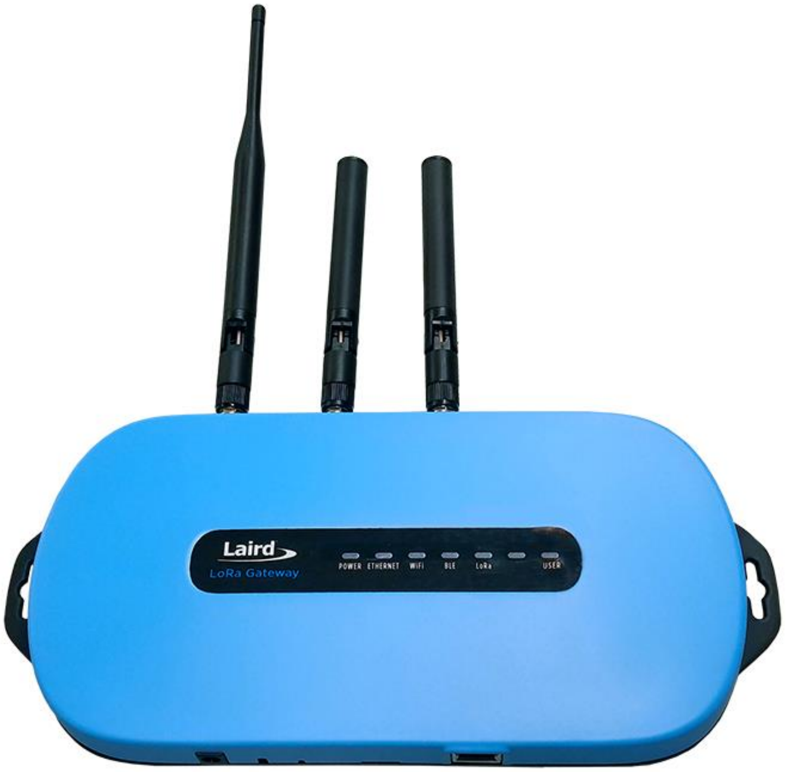


Figure 1: Sentrius RG1XX Gateway

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. Laird Connectivity 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 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.”

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 [X].

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
- 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].

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
- 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 up 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*.

4.3 Add the Gateway to AWS IoT

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

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 (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

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 Core console.
- Click **Wireless connectivity** in the navigation panel on the left.
- Click **Intro**, and then click Get started. This step is needed to pre-populate the default profiles.
- Under Add LoRaWAN gateways and wireless devices, 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.

4.4 Add a LoRaWAN Device to AWS IoT

4.4.1 Preparation

Locate and note the following specifications about your endpoint device.

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, NwkSkey, FNwkSIntKey, SNwkSIntKey, AppSKey

4.4.2 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 **Wireless connectivity**.
- In the navigation pane, click **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.
  - UlRate60
  - UlBucketSize4096
  - DlRate60
  - DlBucketSize4096
  - AddGwMetadatatrue
  - DevStatusReqFreq24
  - DrMax15
  - TargetPer5
  - MinGwDiversity1

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 [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 **Wireless connectivity**, and then **Destinations**.
- Click **Add Destination**.
- On the Add destination page, in the **Permissions** section, for the **IAM Role**, select the IAM Role created in section 4.2.2 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.
- After entering 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.
- Set the Rule Query Statement as follows:

```
SELECT * FROM ‘iot/topic’
```

- Click **Add Action** in the *Set one or more actions* section.
- Select **Republish a message to an 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 2 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 6.

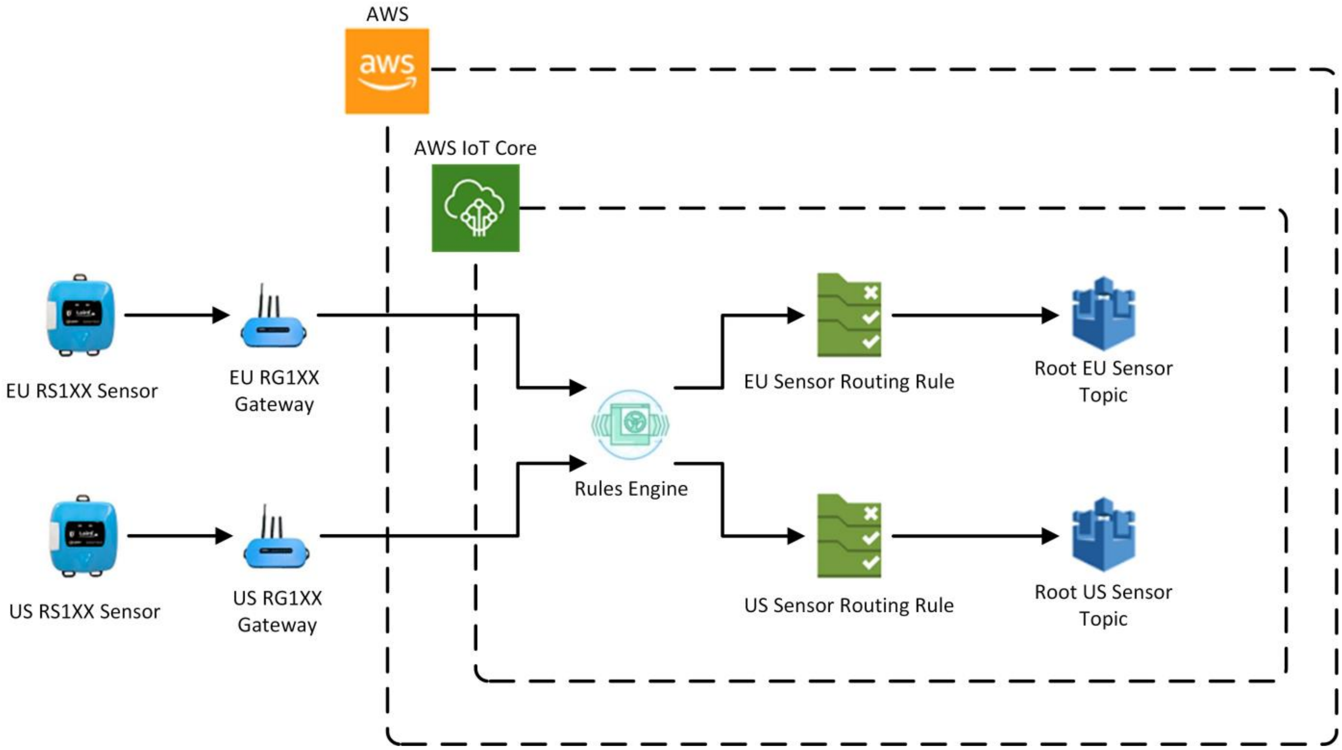


Figure 2: Root MQTT Topic via Destination Rule

4.4.4 Register the Device

- Go to the AWS IoT Core console.
- Click **Wireless connectivity** 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 4.4.2 Verify Device & Service Profiles).

- Click **Next**.
- Choose the destination you created earlier (ProcessLoRa) 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.

5 Set up the Gateway

5.1 Set up hardware

The following describes the steps required to setup the RG1xx Gateway. Figure 3 shows the hardware features of the gateway.

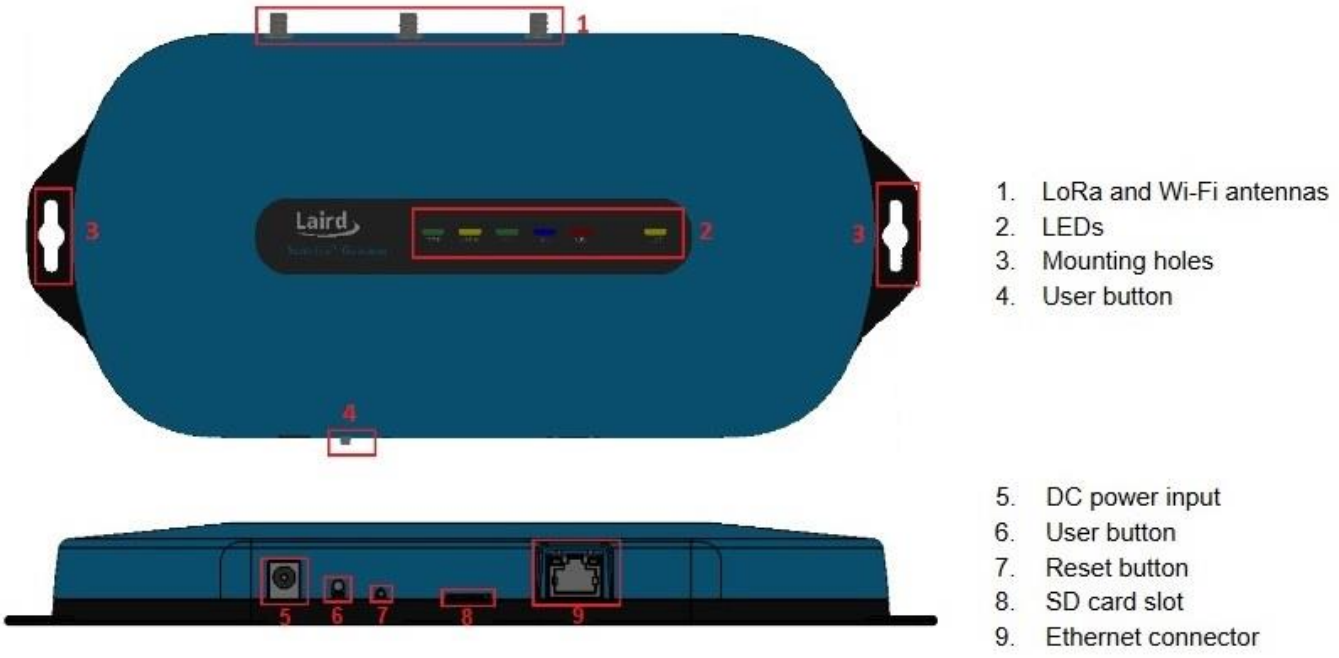


Figure 3: Sentrius RG1xx Gateway hardware features

5.1.1 Physical Connectivity

The supplied antennae are first connected to the gateway before power-up. Figure 4 indicates the location of the LoRa and Wi-Fi antennae.

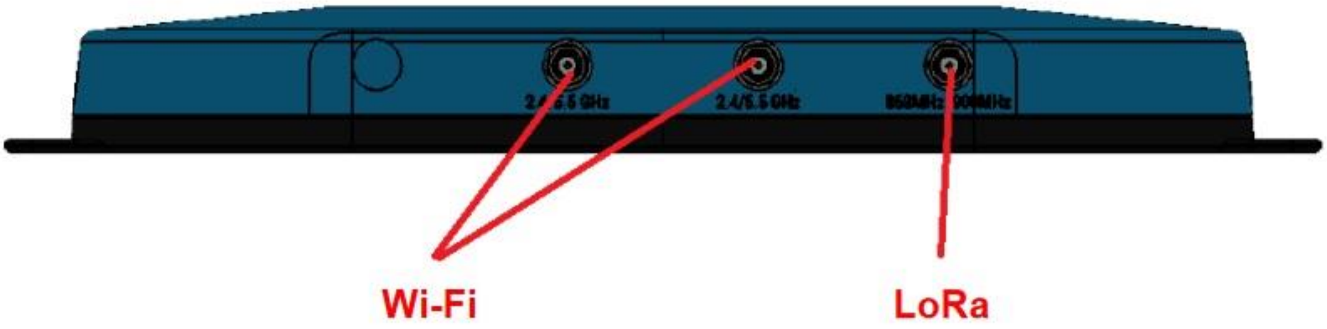


Figure 4: RG1xx antennae connectivity

As shown in Figure 5, 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).

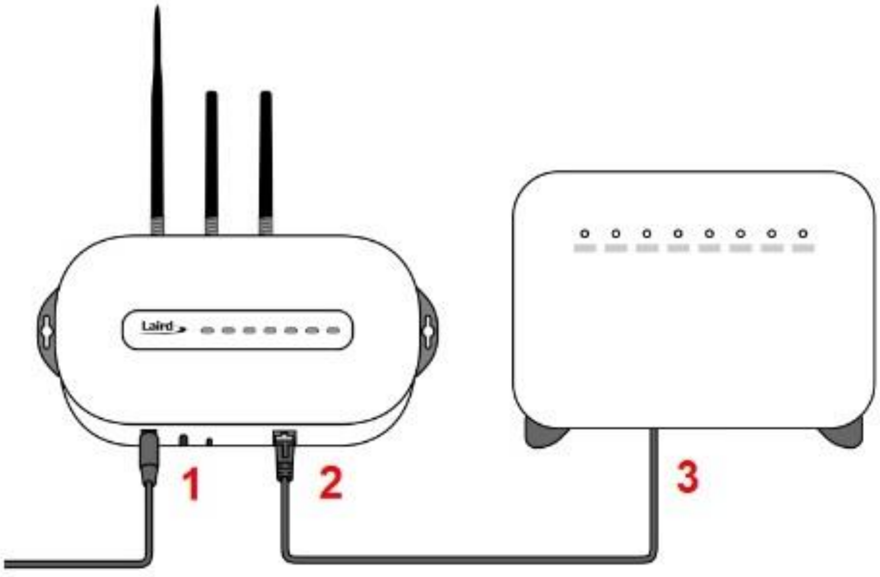


Figure 5: Gateway physical connectivity

5.1.2 Gateway LEDs

The LED array visible on the front panel of the RG1XX gateway is shown in Figure 6. Table 1 describes the purpose of each LED.

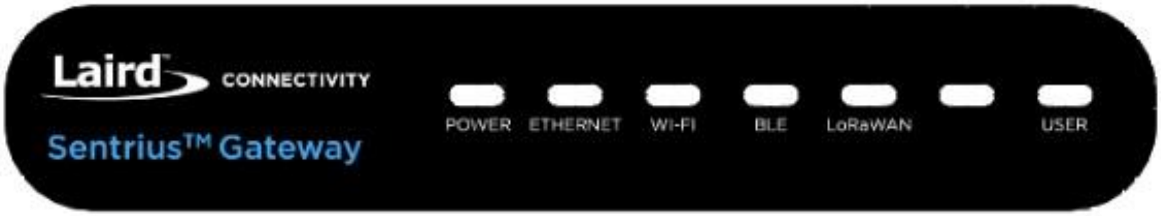


Figure 6: RG1XX LEDs

Table 1: RG1XX LEDs

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

5.1.3 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 7 with the last three bytes highlighted.

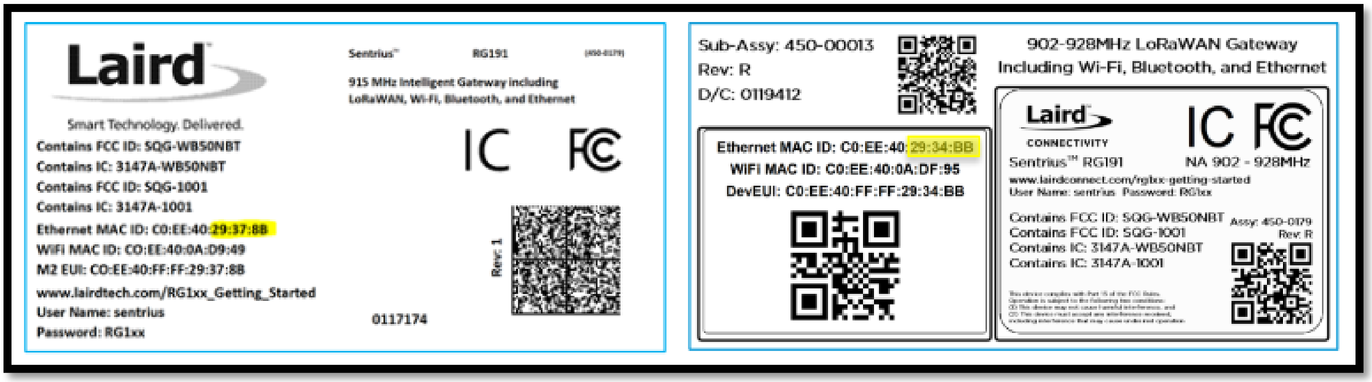
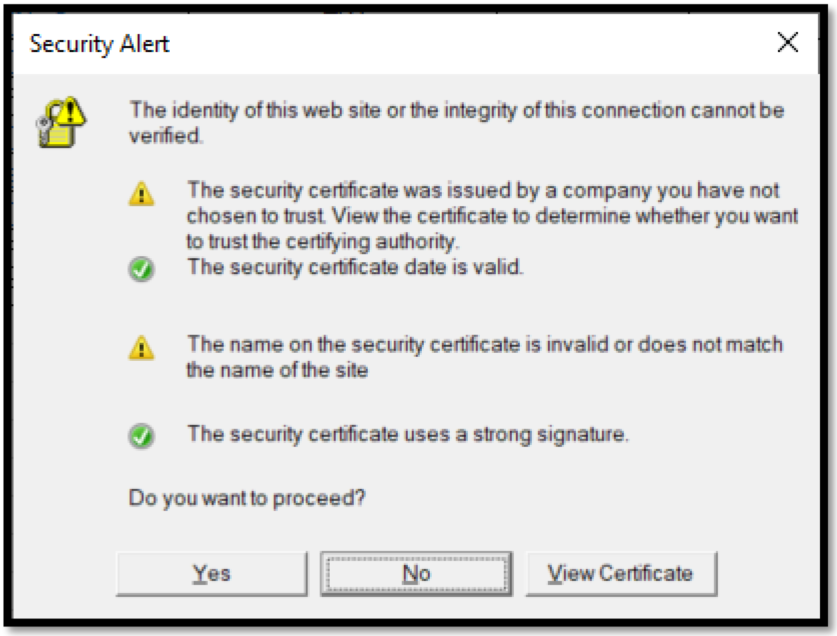

Laird
Smart Technology. Delivered.
Sentrius™
RG191
915 MHz Intelligent Gateway including LoRaWAN, Wi-Fi, Bluetooth, and Ethernet
Contains FCC ID: SQG-WB50NBT
Contains IC: 3147A-WB50NBT
Contains FCC ID: SQG-1001
Contains IC: 3147A-1001
Ethernet MAC ID: C0:EE:40:29:37:8B
WiFi MAC ID: C0:EE:40:0A:D9:49
M2 EUI: C0:EE:40:FF:FF:29:37:8B
www.lairdtech.com/RG1xx_Getting_Started
User Name: sentrius
Password: RG1xx
0117174
Rev: 1
Sub-Assy: 450-00013
Rev: R
D/C: 0119412
Ethernet MAC ID: C0:EE:40:29:34:BB
WIFI MAC ID: C0:EE:40:0A:DF:95
DevEUI: C0:EE:40:FF:FF:29:34:BB
902-928MHz LoRaWAN Gateway Including Wi-Fi, Bluetooth, and Ethernet
Laird CONNECTIVITY
Sentrius™ RG191
NA 902 - 928MHz
www.lairdconnect.com/rg1xx-getting-started
User Name: sentrius Password: RG1xx
Contains FCC ID: SQG-WB50NBT
Contains FCC ID: SQG-1001
Contains IC: 3147A-WB50NBT
Contains IC: 3147A-1001
Assy: 450-0179
Rev: R

Figure 7: 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 8 is first shown. Click **Yes** to proceed.



Security Alert

The identity of this web site or the integrity of this connection cannot be verified.

The security certificate was issued by a company you have not chosen to trust. View the certificate to determine whether you want to trust the certifying authority.

The security certificate date is valid.

The name on the security certificate is invalid or does not match the name of the site

The security certificate uses a strong signature.

Do you want to proceed?

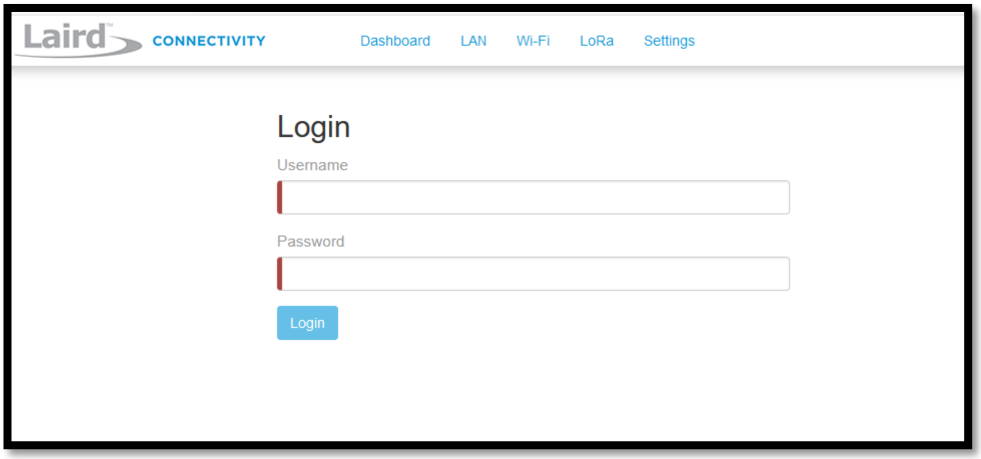
Figure 8: Dialog shown when first opening gateway web interface

The gateway web interface log in page appears as shown in Figure 9. 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**.



Login

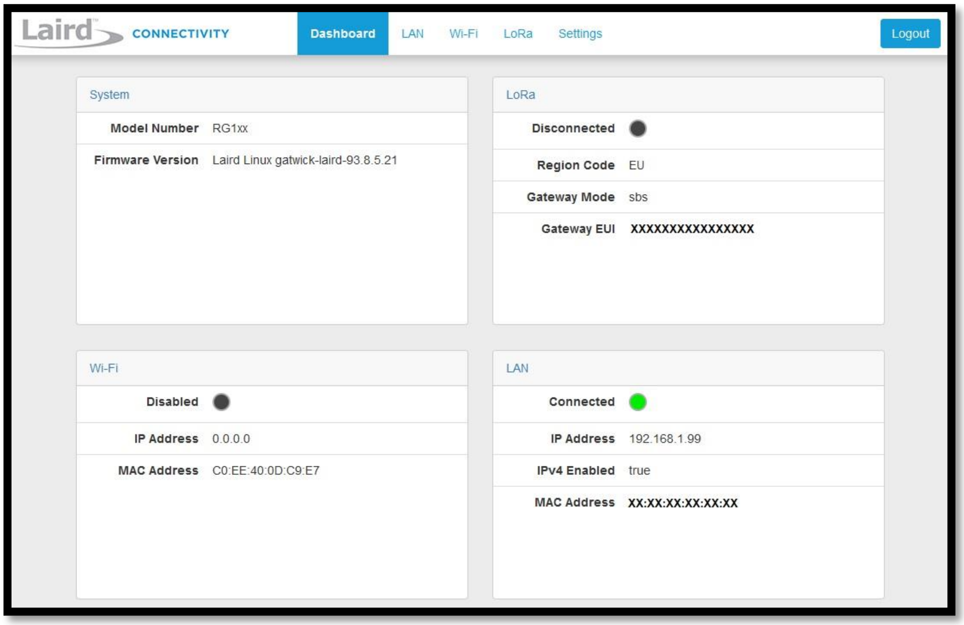
Username

Password

Figure 9: Gateway web interface log in page

The gateway dashboard appears as shown in Figure 10. 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



System

Model Number	RG1xx
Firmware Version	Laird Linux gatwick-laird-93.8.5.21

LoRa

Disconnected	●
Region Code	EU
Gateway Mode	sbs
Gateway EUI	XXXXXXXXXXXXXXXX

Wi-Fi

Disabled	●
IP Address	0.0.0.0
MAC Address	C0:EE:40:0D:C9:E7

LAN

Connected	●
IP Address	192.168.1.99
IPv4 Enabled	true
MAC Address	XX:XX:XX:XX:XX:XX

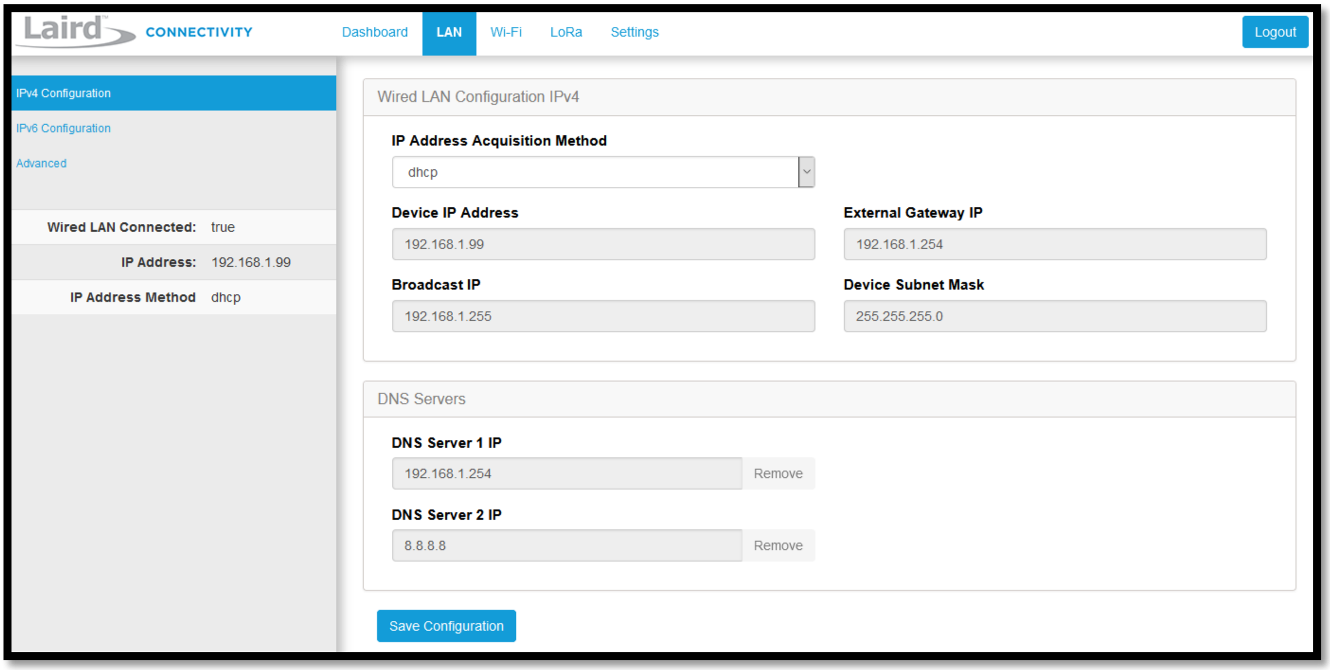
Figure 10: Gateway Dashboard page

5.1.4 Ethernet setup

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

5.1.4.1 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 11.



Wired LAN Connected: true
IP Address: 192.168.1.99
IP Address Method dhcp

IP Address Acquisition Method: dhcp
Device IP Address: 192.168.1.99
External Gateway IP: 192.168.1.254
Broadcast IP: 192.168.1.255
Device Subnet Mask: 255.255.255.0

DNS Server 1 IP: 192.168.1.254
DNS Server 2 IP: 8.8.8.8

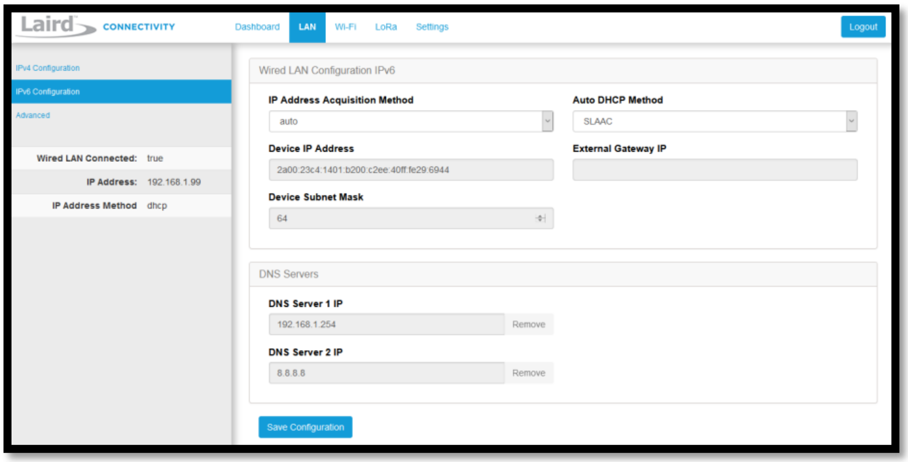
Figure 11: Gateway IPv4 configuration

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.

5.1.4.2 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 12.



Wired LAN Connected:	true
IP Address:	192.168.1.99
IP Address Method	dhcp

Wired LAN Configuration IPv6

IP Address Acquisition Method: auto

Auto DHCP Method: SLAAC

Device IP Address: 2a00:23c4:1401:b200:c2ee:40ff:fe29:6944

External Gateway IP:

Device Subnet Mask: 64

DNS Servers

DNS Server 1 IP: 192.168.1.254

DNS Server 2 IP: 8.8.8.8

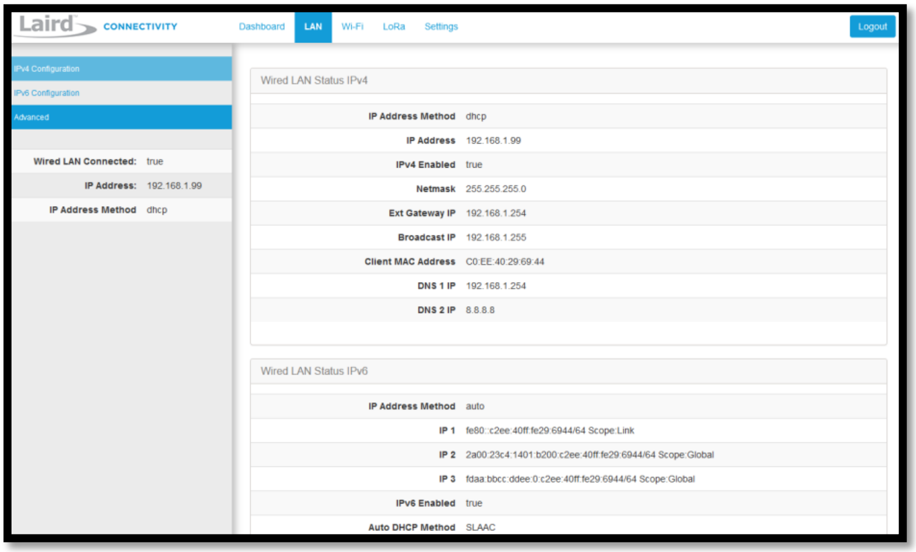
Figure 12: Gateway IPv6 configuration

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

5.1.4.3 Advanced page

From the LAN page, clicking **Advanced** in the left submenu. IPv4 and IPv6 information appears as shown in Figure 13.



Wired LAN Connected:	true
IP Address:	192.168.1.99
IP Address Method	dhcp

Wired LAN Status IPv4

IP Address Method	dhcp
IP Address	192.168.1.99
IPv4 Enabled	true
Netmask	255.255.255.0
Ext Gateway IP	192.168.1.254
Broadcast IP	192.168.1.255
Client MAC Address	C0:EE:40:29:69:44
DNS 1 IP	192.168.1.254
DNS 2 IP	8.8.8.8

Wired LAN Status IPv6

IP Address Method	auto
IP 1	fe80::c2ee:40ff:fe29:6944/64 Scope:Link
IP 2	2a00:23c4:1401:b200:c2ee:40ff:fe29:6944/64 Scope:Global
IP 3	fdaa:bbcc:ddee:0:c2ee:40ff:fe29:6944/64 Scope:Global
IPv6 Enabled	true
Auto DHCP Method	SLAAC

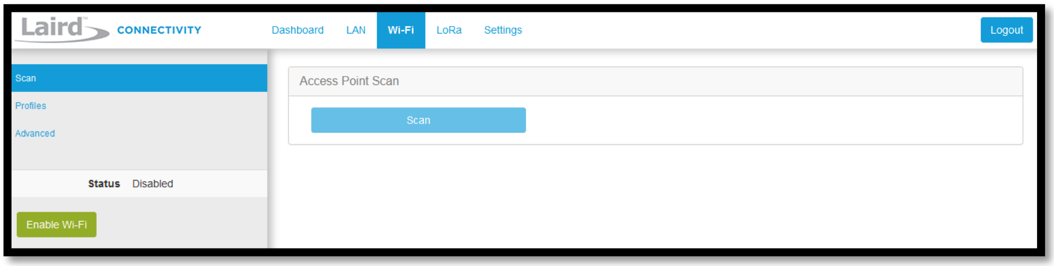
Figure 13: LAN Advanced page

5.1.5 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.

5.1.5.1 Adding an access point

Click **Wi-Fi** in the top menu. The Wi-Fi page appears as shown in Figure 14.

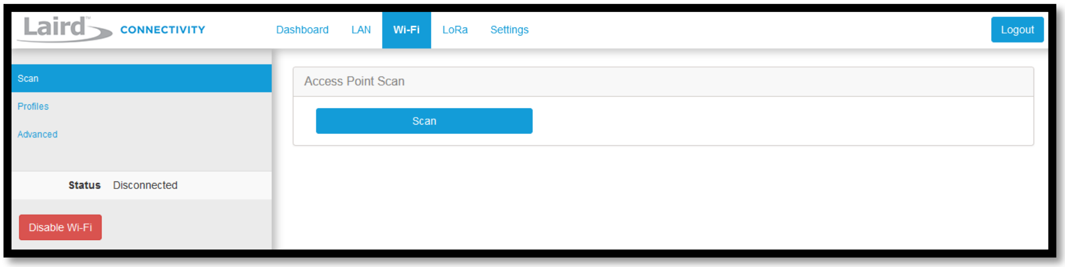


Access Point Scan

Status Disabled

Figure 14: 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 lluminates steadily. *Enable Wi-Fi* updates to display *Disable Wi-Fi* when Wi-Fi is active, as shown in Figure 15.

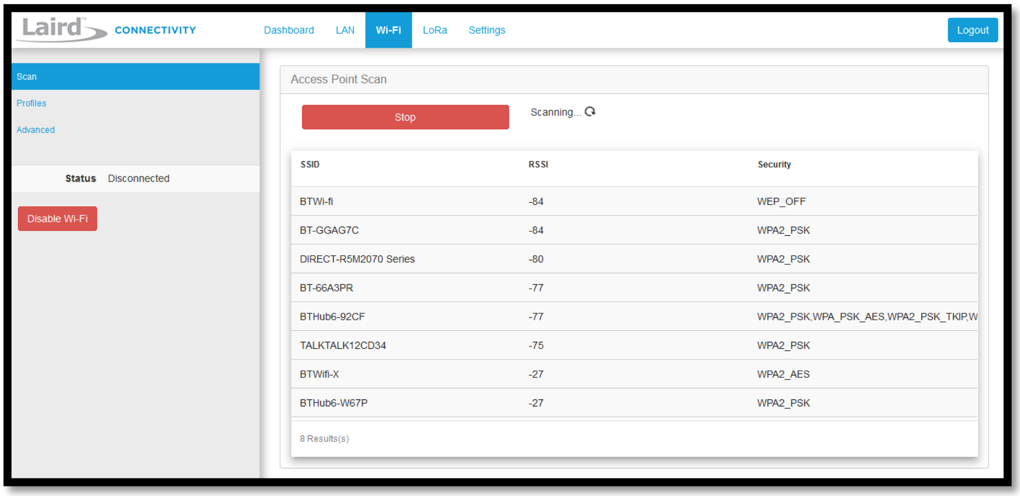


Access Point Scan

Status Disconnected

Figure 15: 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 16.



Access Point Scan

Status Disconnected

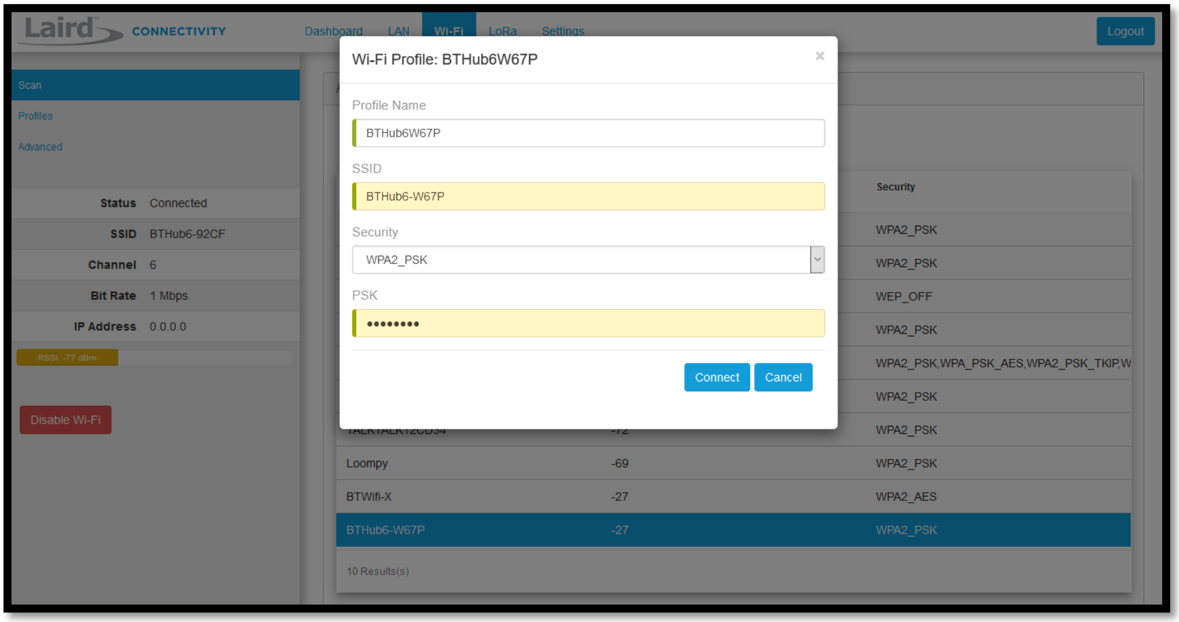
Scanning...

SSID	RSSI	Security
BTWi-fi	-84	WEP_OFF
BT-GGAG7C	-84	WPA2_PSK
DIRECT-R5M2070 Series	-80	WPA2_PSK
BT-66A3PR	-77	WPA2_PSK
BTHub6-92CF	-77	WPA2_PSK,WPA_PSK_AES,WPA2_PSK_TKIP,W
TALKTALK12CD34	-75	WPA2_PSK
BTWifi-X	-27	WPA2_AES
BTHub6-W67P	-27	WPA2_PSK

8 Results(s)

Figure 16: Access Point Scan results

Click your desired access point. Enter credentials as shown in Figure 17.



Status	Connected
SSID	BTHub6-92CF
Channel	6
Bit Rate	1 Mbps
IP Address	0.0.0.0

Wi-Fi Profile: BTHub6W67P

Profile Name: BTHub6W67P

SSID: BTHub6-W67P

Security: WPA2_PSK

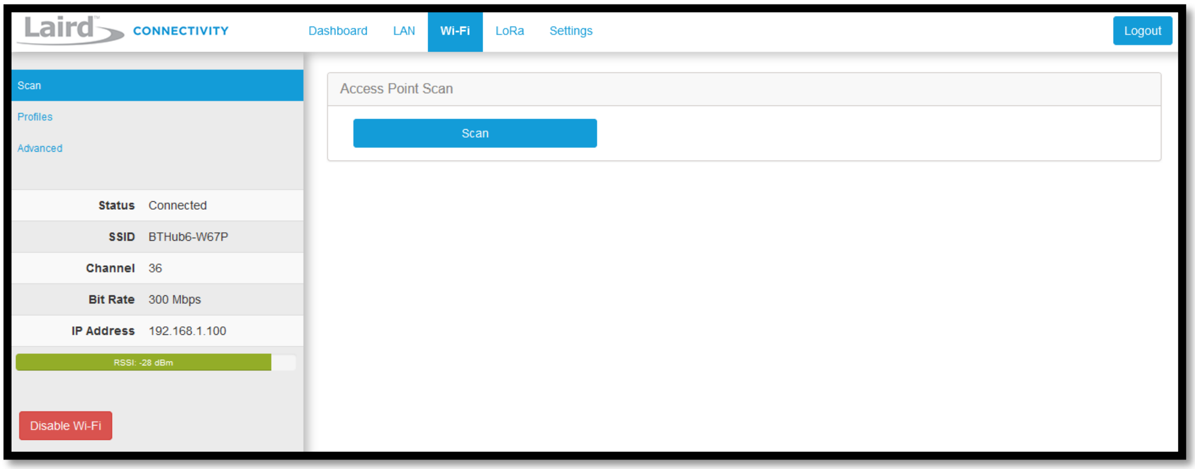
PSK: ••••••••

		Security
		WPA2_PSK
		WPA2_PSK
		WEP_OFF
		WPA2_PSK
		WPA2_PSK,WPA_PSK_AES,WPA2_PSK_TKIP,W
		WPA2_PSK
TALKTALK12CD34	-72	WPA2_PSK
Loompy	-69	WPA2_PSK
BTWifi-X	-27	WPA2_AES
BTHub6-W67P	-27	WPA2_PSK

10 Results(s)

Figure 17: Access Point details

Click **Connect** to connect to the access point. Figure 18 shows the updated Wi-Fi page.

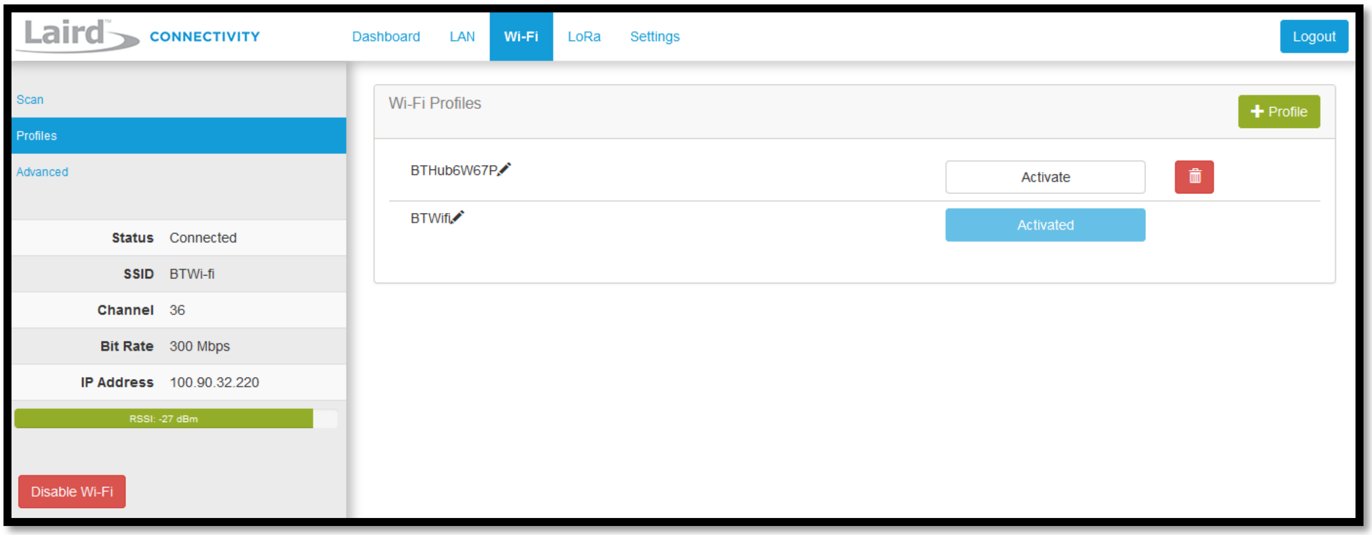


Status	Connected
SSID	BTHub6-W67P
Channel	36
Bit Rate	300 Mbps
IP Address	192.168.1.100

Figure 18: Successful connection to access point

5.1.5.2 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 19.



Wi-Fi Profiles

BTHub6W67P

BTWifi

Status	Connected
SSID	BTWi-fi
Channel	36
Bit Rate	300 Mbps
IP Address	100.90.32.220

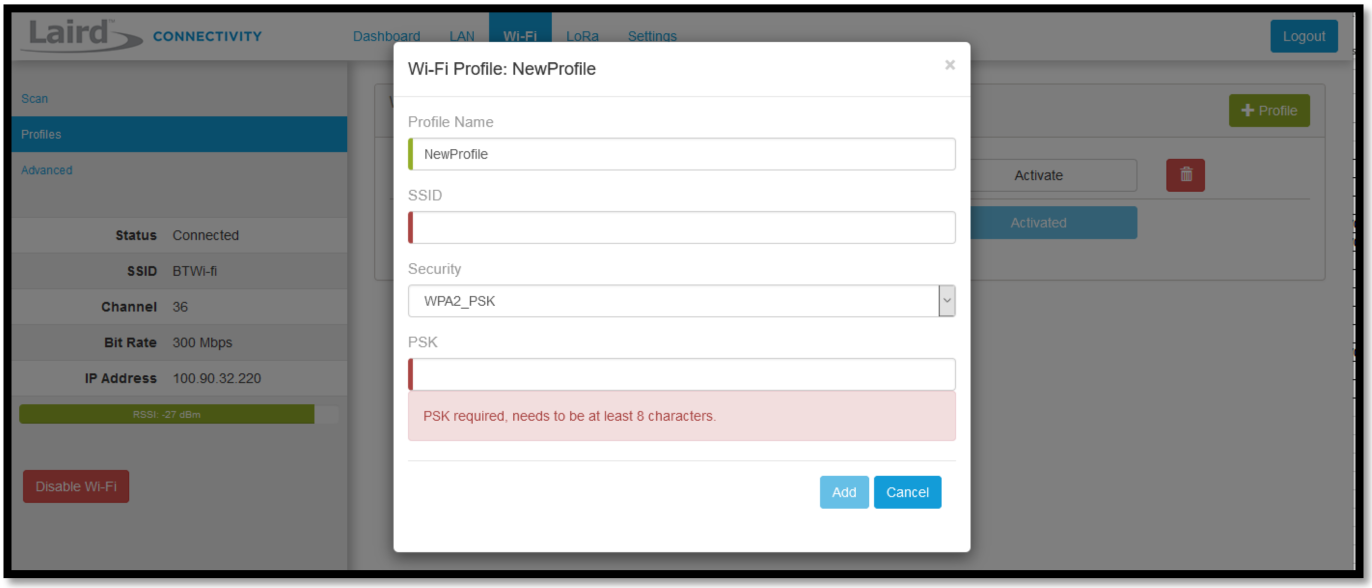
RSSI: -27 dBm

Figure 19: Wi-Fi Profiles page

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

5.1.5.3 Manually adding a profile

Manually add an access point by clicking **+ Profile** as shown in Figure 20. Then click **Add** to activate the new profile.



Wi-Fi Profile: NewProfile

Profile Name: NewProfile

SSID:

Security: WPA2_PSK

PSK:

PSK required, needs to be at least 8 characters.

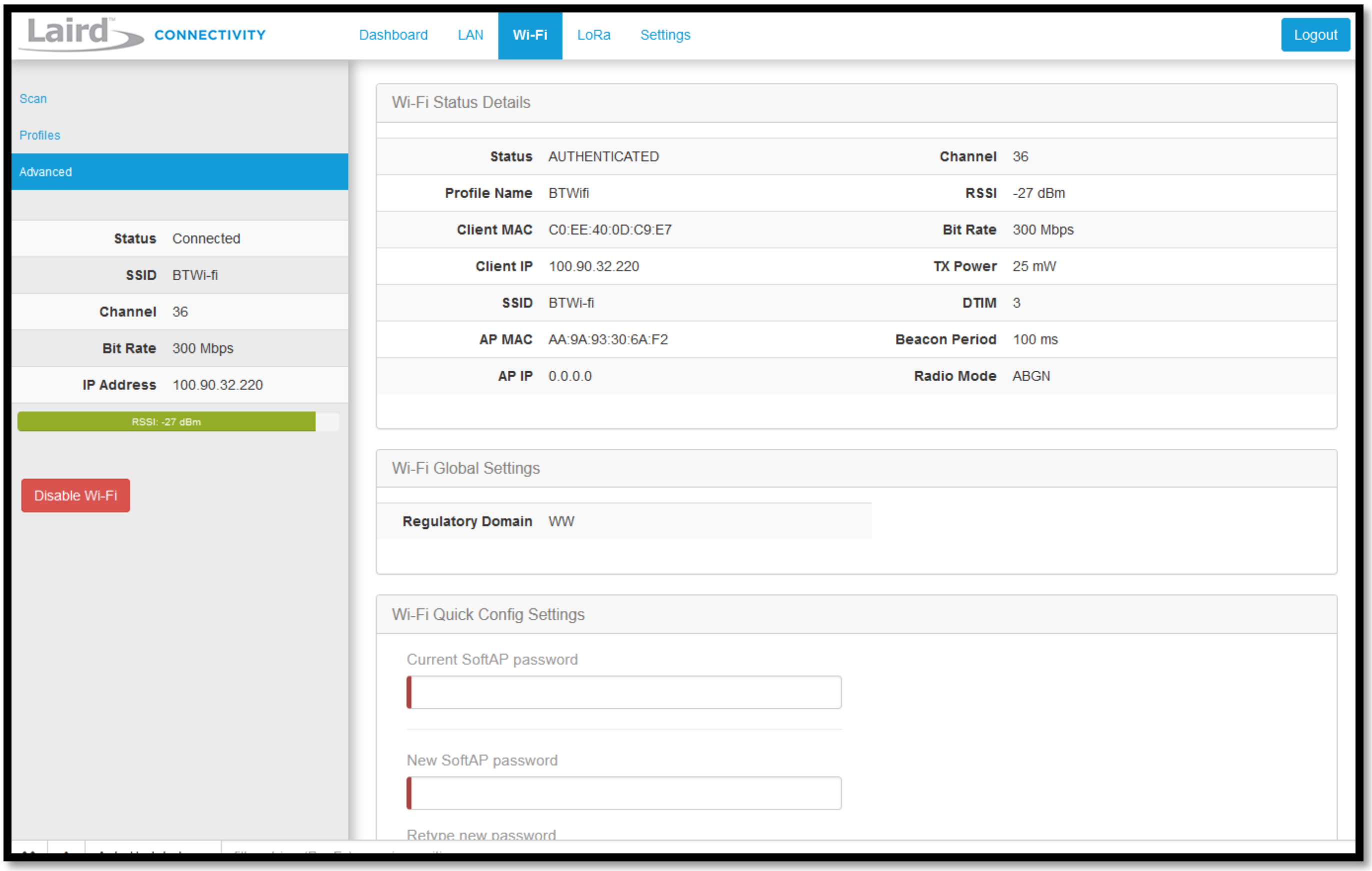
Status	Connected
SSID	BTWi-fi
Channel	36
Bit Rate	300 Mbps
IP Address	100.90.32.220

RSSI: -27 dBm

Figure 20: Manually adding an access point profile

5.1.5.4 Advanced page

Click **Advanced** in the submenu on the left to open Advanced page as shown in Figure 21. This page displays the parameters of the current access point.



Status	Connected
SSID	BTWi-fi
Channel	36
Bit Rate	300 Mbps
IP Address	100.90.32.220

Wi-Fi Status Details

Status	AUTHENTICATED	**Channel**	36
Profile Name	BTWifi	**RSSI**	-27 dBm
Client MAC	C0:EE:40:0D:C9:E7	**Bit Rate**	300 Mbps
Client IP	100.90.32.220	**TX Power**	25 mW
SSID	BTWi-fi	**DTIM**	3
AP MAC	AA:9A:93:30:6A:F2	**Beacon Period**	100 ms
AP IP	0.0.0.0	**Radio Mode**	ABGN

Wi-Fi Global Settings

Regulatory Domain	WW

Wi-Fi Quick Config Settings

Current SoftAP password

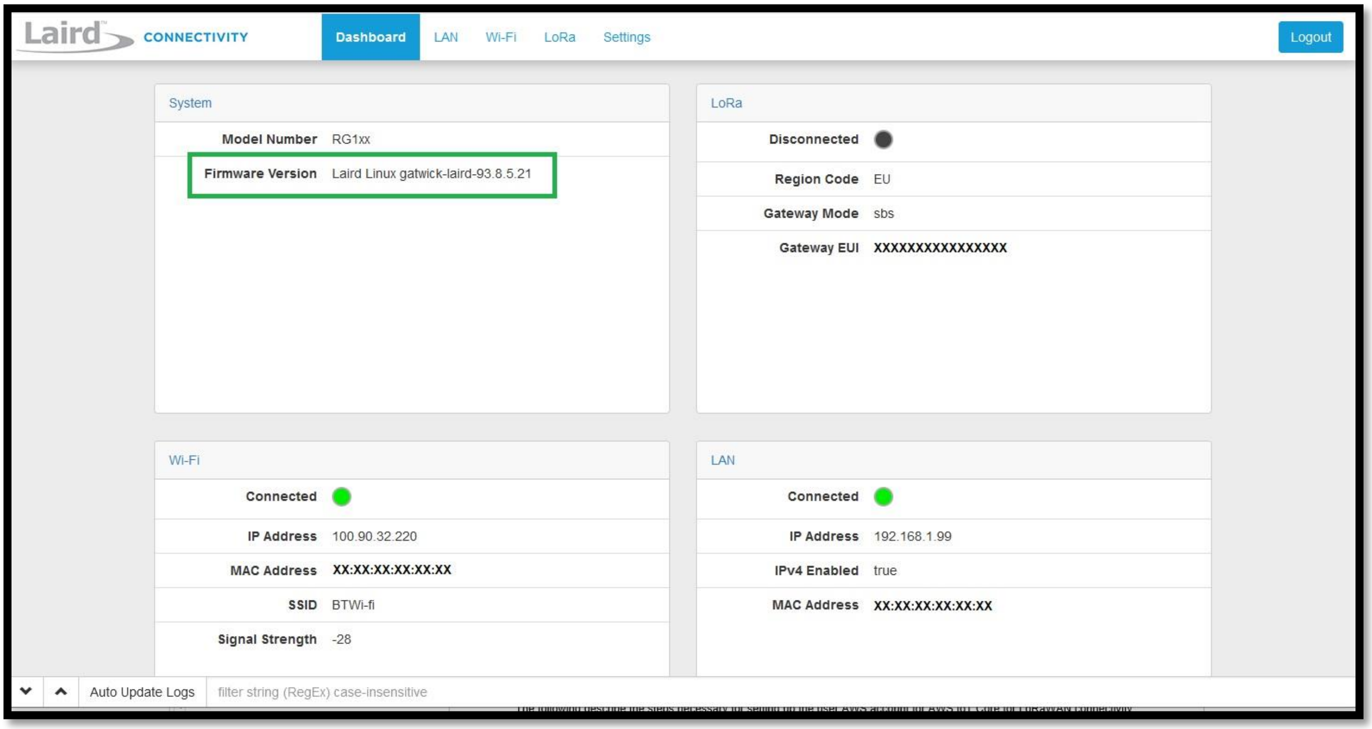
New SoftAP password

Retype new password

Figure 21: Wi-Fi Advanced page

5.2 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 can be verified from the web interface Dashboard as shown in Figure 22.



System

Model Number	RG1xx
Firmware Version	Laird Linux gatwick-laird-93.8.5.21

LoRa

Disconnected	●
Region Code	EU
Gateway Mode	sbs
Gateway EUI	XXXXXXXXXXXXXXXX

Wi-Fi

Connected	●
IP Address	100.90.32.220
MAC Address	XX:XX:XX:XX:XX:XX
SSID	BTWi-fi
Signal Strength	-28

LAN

Connected	●
IP Address	192.168.1.99
IPv4 Enabled	true
MAC Address	XX:XX:XX:XX:XX:XX

The following describe the steps necessary for setting up the user AWS account for AWS IoT Core for LoRaWAN connectivity

Figure 22: Verifying the gateway firmware version

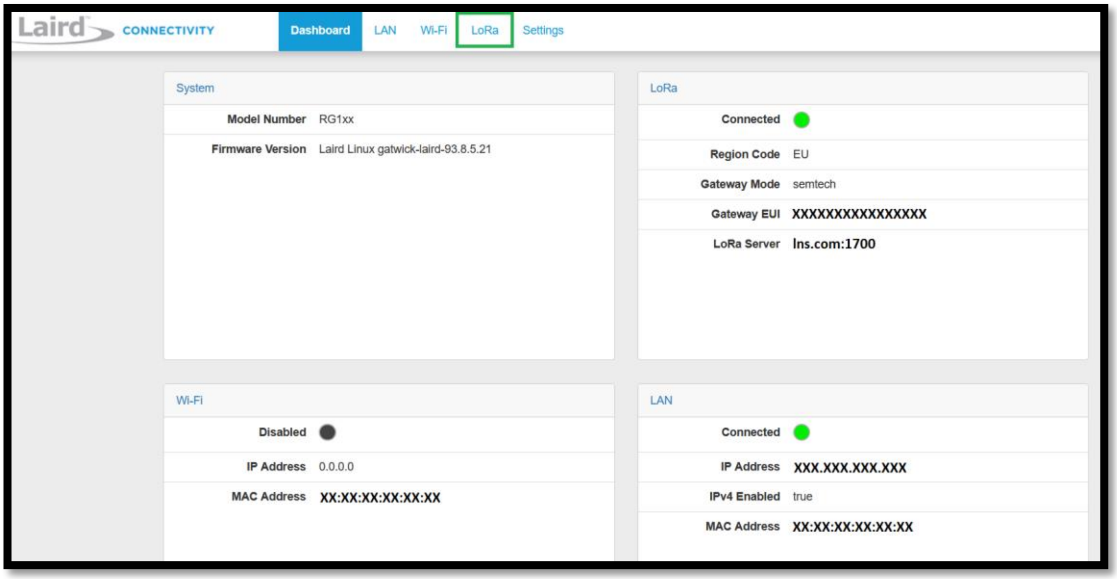
If the firmware version is prior to 93.8.5.25, you must upgrade as shown in the Gateway OTA Updates section.

5.3 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.

5.3.1 Enabling the Basics Station Packet Forwarder

Click **LoRa** in the main menu as shown in Figure 23.



System

Model Number RG1xx
Firmware Version Laird Linux gatwick-laird-93.8.5.21

LoRa

Connected ●
Region Code EU
Gateway Mode semtech
Gateway EUI XXXXXXXXXXXXXXXX
LoRa Server lns.com:1700

Wi-Fi

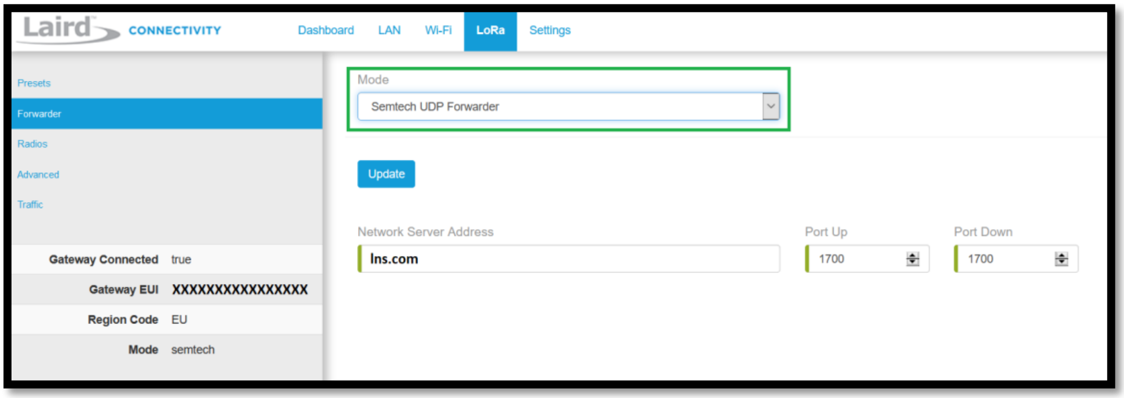
Disabled ●
IP Address 0.0.0.0
MAC Address XX:XX:XX:XX:XX:XX

LAN

Connected ●
IP Address XXX.XXX.XXX.XXX
IPv4 Enabled true
MAC Address XX:XX:XX:XX:XX:XX

Figure 23: 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 24.



Mode: Semtech UDP Forwarder

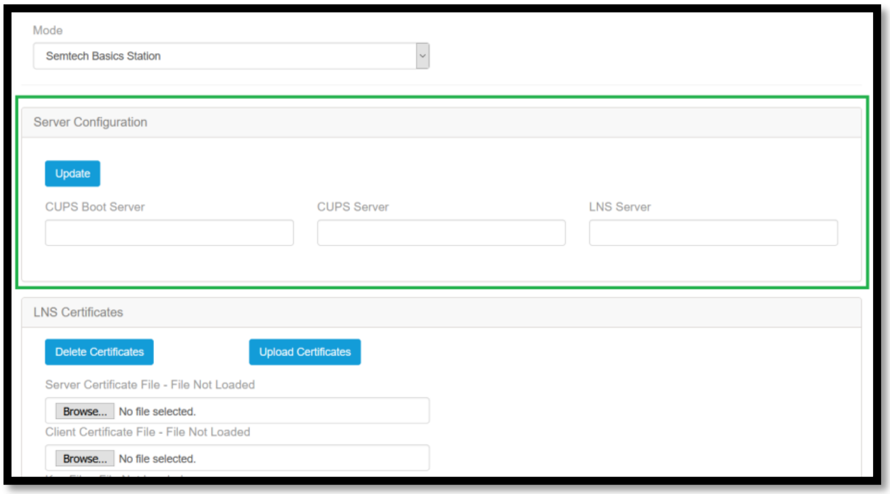
Network Server Address: lns.com
Port Up: 1700
Port Down: 1700

Gateway Connected true
Gateway EUI XXXXXXXXXXXXXXXX
Region Code EU
Mode semtech

Figure 24: Packet Forwarder selection

5.3.2 Configuring end points

The Basics Station configuration page appears as shown in Figure 25. Configure details of the CUPS and LRC endpoints in the *Server Configuration* group.



Mode: Semtech Basics Station

Server Configuration
Update
CUPS Boot Server
CUPS Server
LNS Server
LNS Certificates
Delete Certificates
Upload Certificates
Server Certificate File - File Not Loaded
Browse... No file selected.
Client Certificate File - File Not Loaded
Browse... No file selected.

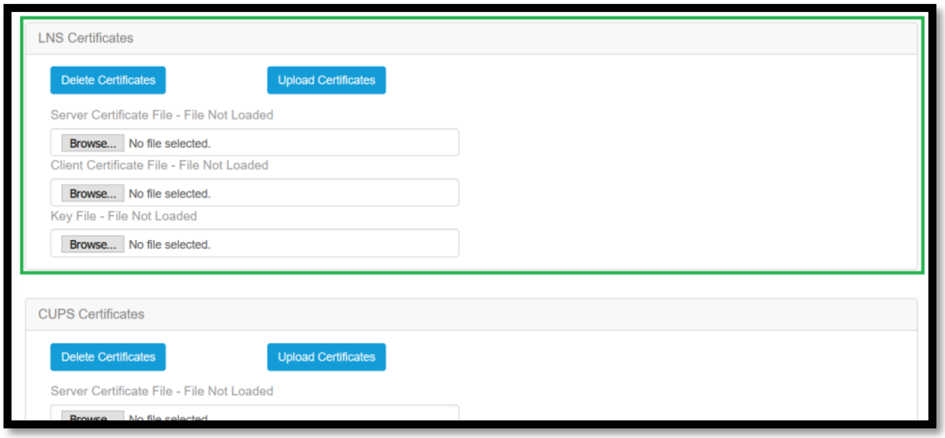
Figure 25: Basics Station Server Configuration

- The ‘CUPS Server’ and ‘CUPS Boot Server’ should be set to the CUPS Endpoint value noted during section 4.3.
- The ‘LNS Server’ should be set to the LNS Endpoint value noted during section 4.3.

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

5.3.3 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 26.



LNS Certificates
Delete Certificates
Upload Certificates
Server Certificate File - File Not Loaded
Browse... No file selected.
Client Certificate File - File Not Loaded
Browse... No file selected.
Key File - File Not Loaded
Browse... No file selected.
CUPS Certificates
Delete Certificates
Upload Certificates
Server Certificate File - File Not Loaded
Browse... No file selected.

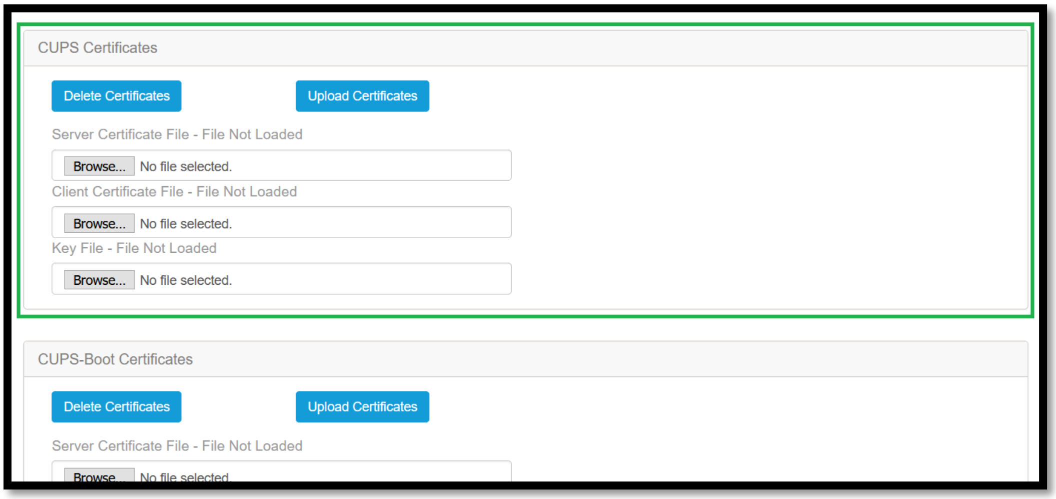
Figure 26: LNS Certificates group

- ‘Server Certificate File’ is the LNS Trust Certificate file stored during the steps described in section 4. This has the .trust file extension, 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 4. This has the .pem extension. Select *All Files (*.*)* should be selected 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 4.

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.

5.3.4 Configuring CUPS certificates

Add details of the CUPS server, if required, via the CUPS Certificates group, as shown in Figure 27.


CUPS Certificates

Server Certificate File - File Not Loaded

No file selected.

Client Certificate File - File Not Loaded

No file selected.

Key File - File Not Loaded

No file selected.

CUPS-Boot Certificates

Server Certificate File - File Not Loaded

No file selected.

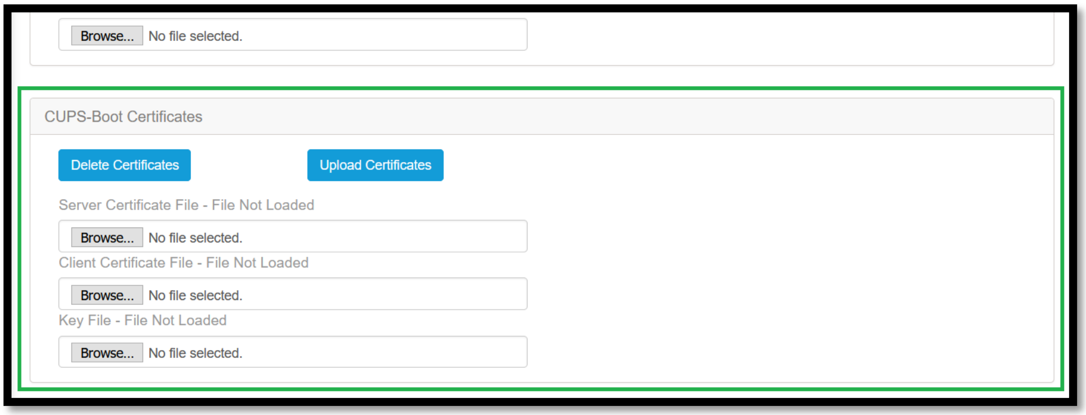
Figure 27: CUPS Certificates group

- The ‘Server Certificate File’ is the CUPS Trust Certificate file stored during the steps described in section 4. This has the .trust file extension, 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 4. 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 4.

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

5.3.5 Configuring CUPS Boot certificates

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


No file selected.

CUPS-Boot Certificates

Server Certificate File - File Not Loaded

No file selected.

Client Certificate File - File Not Loaded

No file selected.

Key File - File Not Loaded

No file selected.

Figure 28: CUPS Boot Certificates group

- The ‘Server Certificate File’ is the CUPS Trust Certificate file stored during the steps described in section 4. This has the .trust file extension, ‘All Files (*.*)’ should be selected 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 4. This has the .pem extension. ‘All Files (*.*)’ should be selected 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 4.

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

5.3.6 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 29. The gateway will restart within a minute, then establish communication with the AWS IoT Core for LoRaWAN instance.

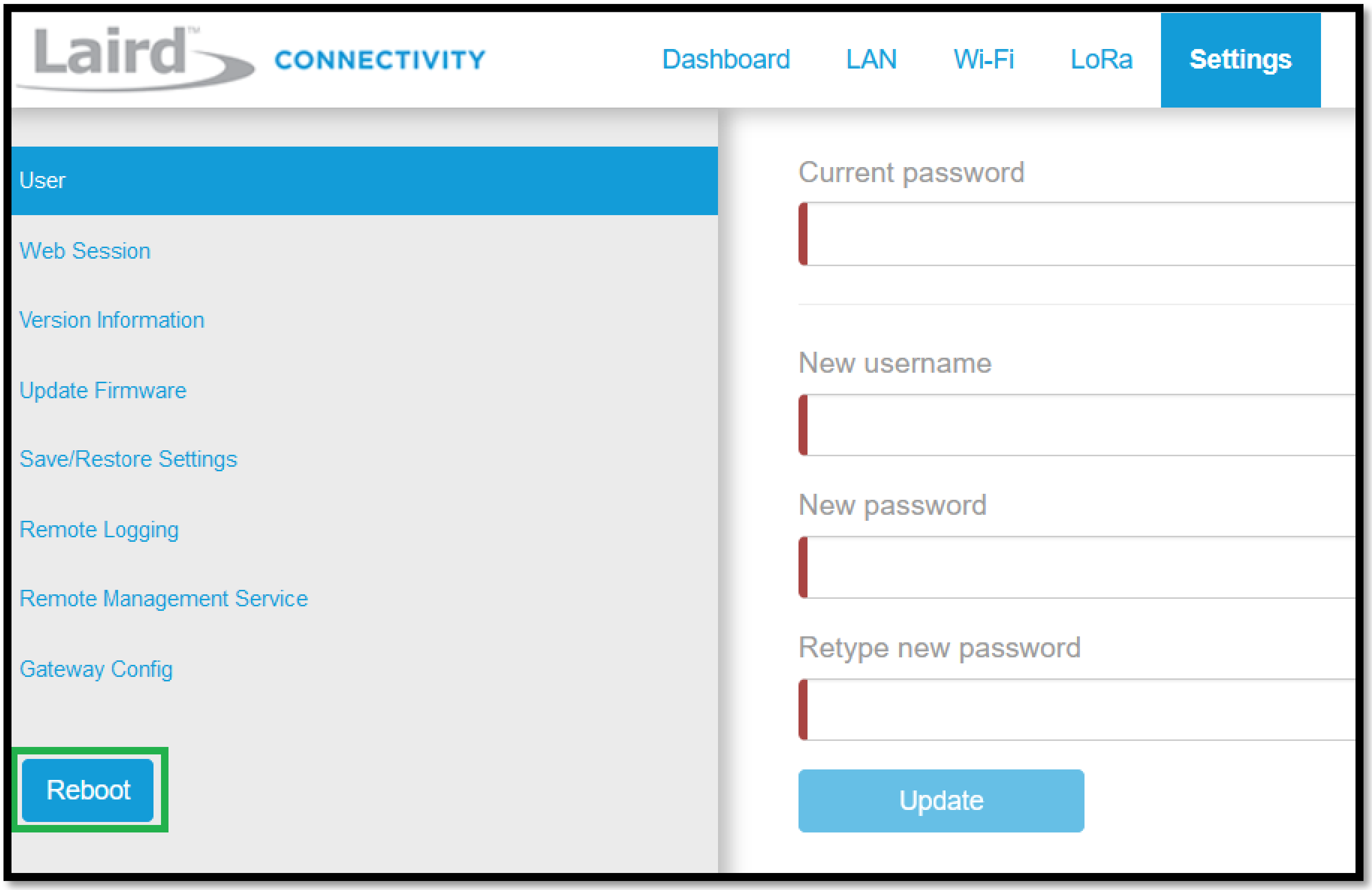


Figure 29: Rebooting the RG1xx

6 Application Examples

The following describe some applications to test the sensor connectivity and demonstrate AWS features. The following are intended for use with Laird's RS1XX range of sensors.

6.1 Laird Connectivity 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 Laird Connectivity 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 30. 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.

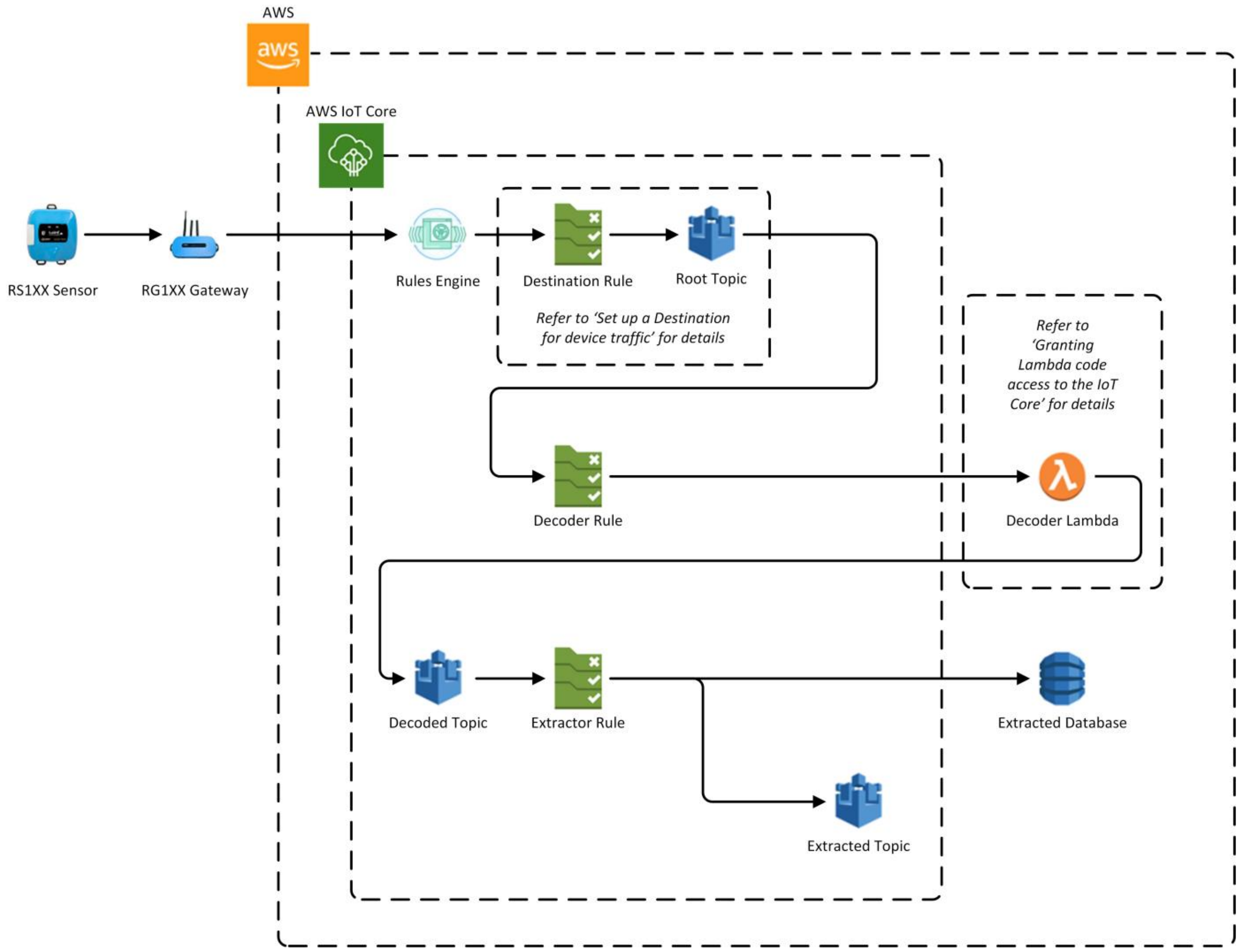


Figure 30: Laird Connectivity Packet Format application architecture

6.1.1 Creating the Decoder Lambda function

First, create the Decoder Lambda function. As shown in Figure 31, click **Lambda** from the AWS landing page main menu.

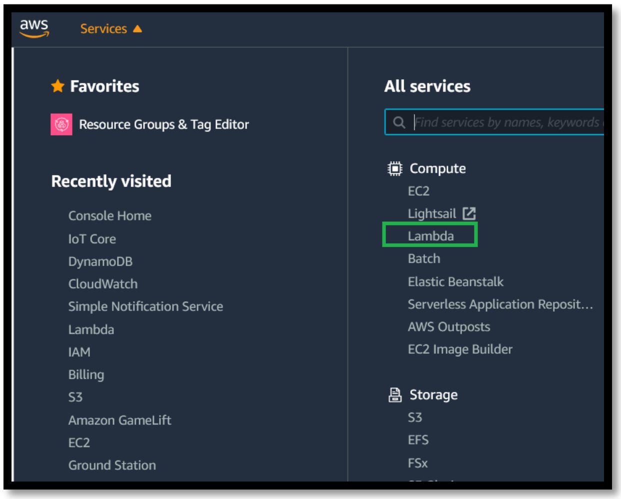


Figure 31: Creating a new Lambda function

This opens the main Lambda page as shown in Figure 32. Click **Create function**.

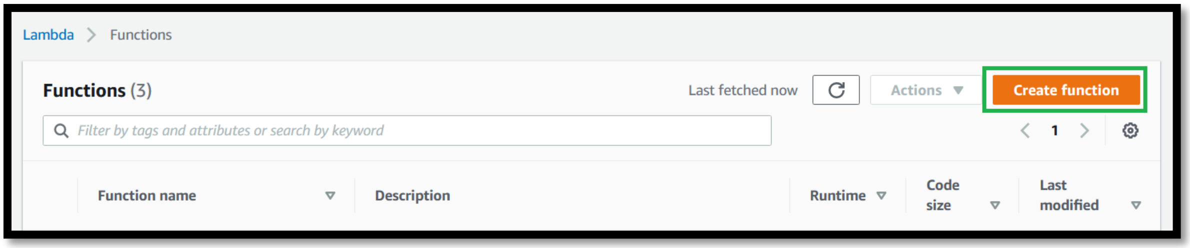
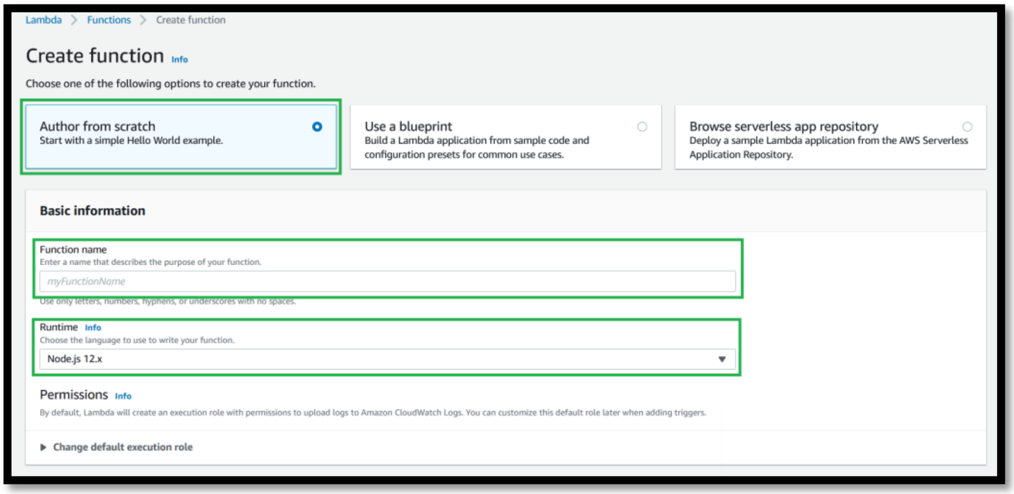


Figure 32: Main Lambda page

This opens the page shown in Figure 33. *Author from scratch* should be selected. Set *Function name* to “Decoder’ and *Runtime* to “NodeJS 10.x”. Click **Create function**.



Choose one of the following options to create your function.

● Author from scratch
Start with a simple Hello World example.

○ Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

○ Browse serverless app repository
Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name
Enter a name that describes the purpose of your function.
myFunctionName
Use only letters, numbers, hyphens, or underscores with no spaces.

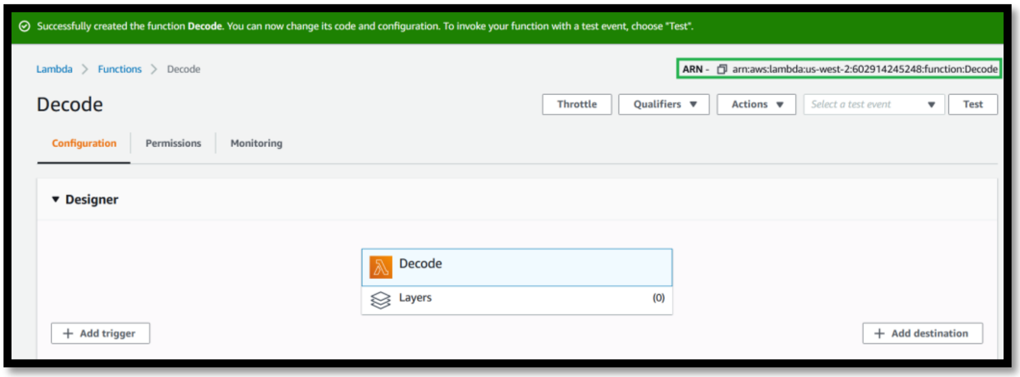
Runtime Info
Choose the language to use to write your function.
Node.js 12.x

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▶ Change default execution role

Figure 33: Setting Lambda function information

The Lambda function designer is as shown in Figure 34. Note the ARN of the Lambda for later use. It is passed to the query used in the Decoder Rule.

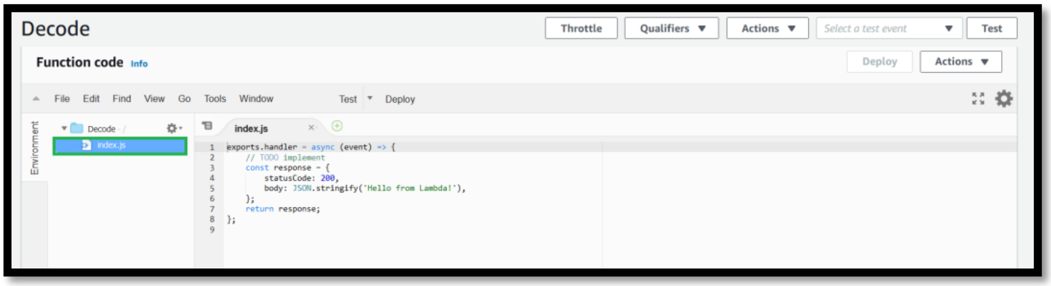


Successfully created the function **Decode**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

ARN - arn:aws:lambda:us-west-2:602914245248:function:Decode

Figure 34: Lambda function designer

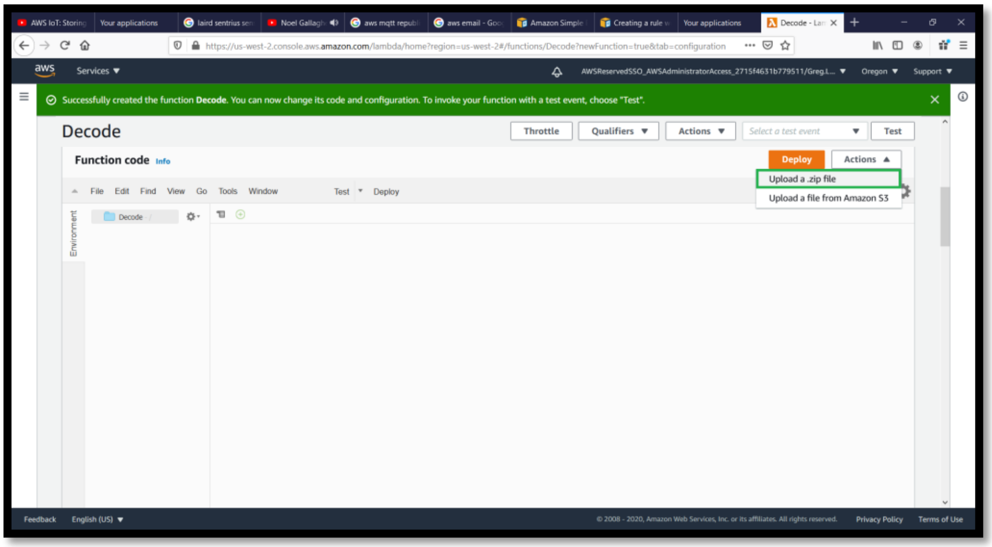
Scroll down to the Lambda function code, as shown in Figure 35. Delete the example file, "index.js." Add the Decoder files either by creating files "library_laird.js", "index.js" and "messages_laird.js" and copying/pasting the content, or by creating a zip file containing the files and uploading to AWS. Note that you must manually create the zip file and it must contain the three files needed in the root directory of the archive.



```
exports.handler = async (event) => {
    // TODO implement
    const response = {
        statusCode: 200,
        body: JSON.stringify('Hello from Lambda!'),
    };
    return response;
};
```

Figure 35: Lambda function body

Figure 36 shows the method where a zip file is uploaded with the example code incorporated in a zip file.

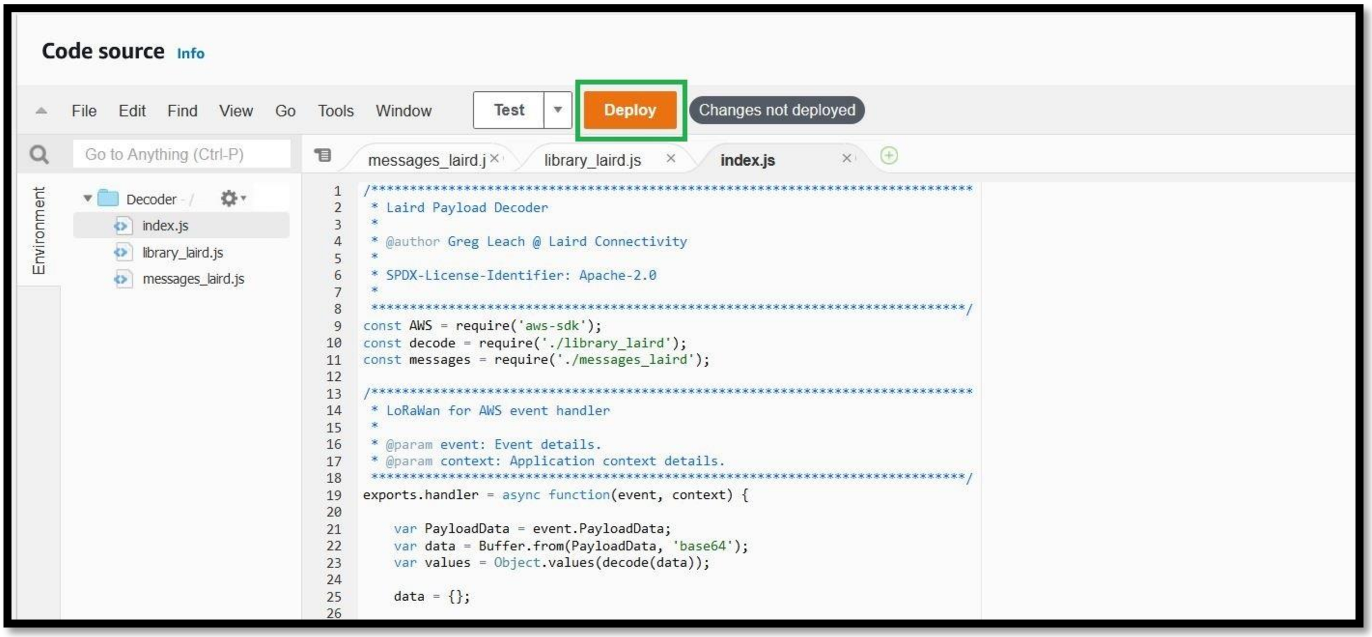


https://us-west-2.console.aws.amazon.com/lambda/home?region=us-west-2#/functions/Decode?newFunction=true&tab=configuration

Successfully created the function **Decode**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Figure 36: Uploading Lambda function content

Once the Lambda code is available, the function content appears as shown in Figure 37. Click **Deploy** to deploy the Lambda code.



```
/*****************************************************************************
 * Laird Payload Decoder
 *
 * @author Greg Leach @ Laird Connectivity
 *
 * SPDX-License-Identifier: Apache-2.0
 *
 ****************************************************************************/
const AWS = require('aws-sdk');
const decode = require('./library_laird');
const messages = require('./messages_laird');

/*****************************************************************************
 * LoRaWan for AWS event handler
 *
 * @param event: Event details.
 * @param context: Application context details.
 ****************************************************************************/
exports.handler = async function(event, context) {

    var PayloadData = event.PayloadData;
    var data = Buffer.from(PayloadData, 'base64');
    var values = Object.values(decode(data));

    data = {};
```

Figure 37: Completed Lambda function code

6.1.2 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 38.

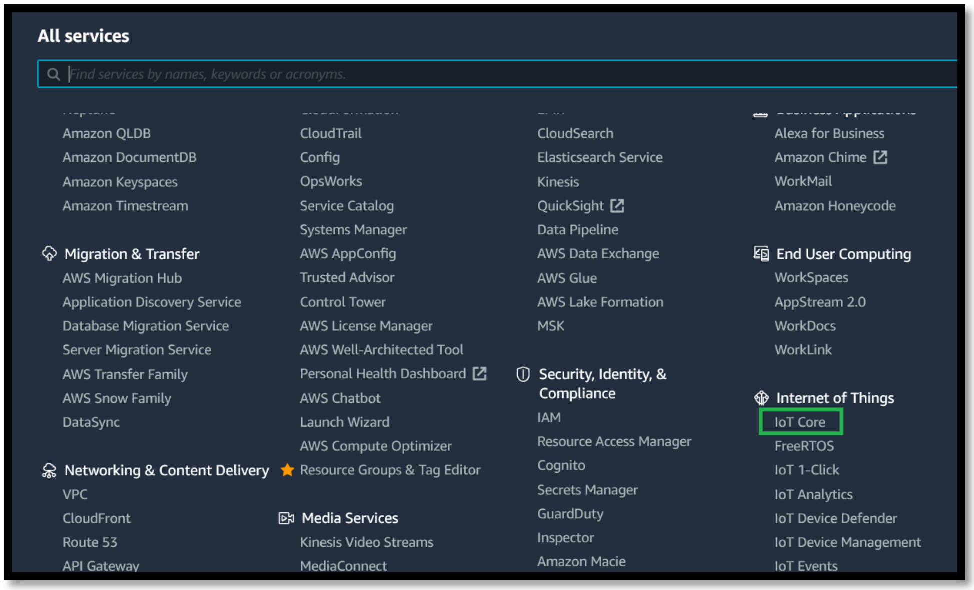


Figure 38: Opening AWS IoT Core

From the side menu, click **Act**, then click **Rules**, as shown in Figure 39. This opens the AWS IoT Core Rules Engine.

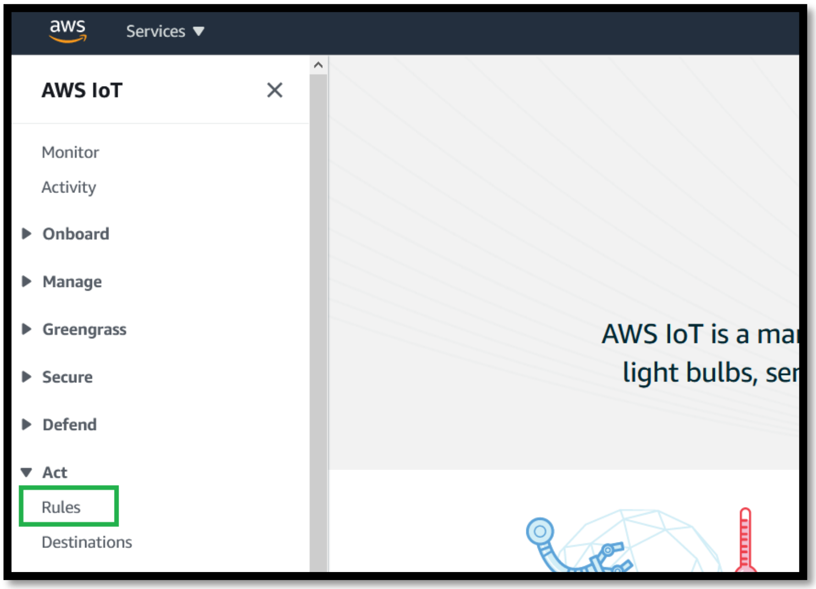


Figure 39: Opening IoT Core Rules Engine

Click **Create** as shown in Figure 40 to create a new Rule.

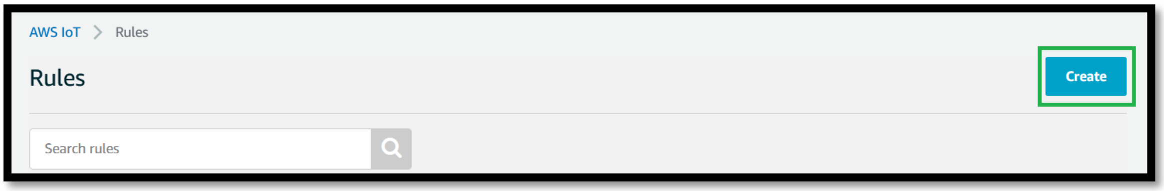
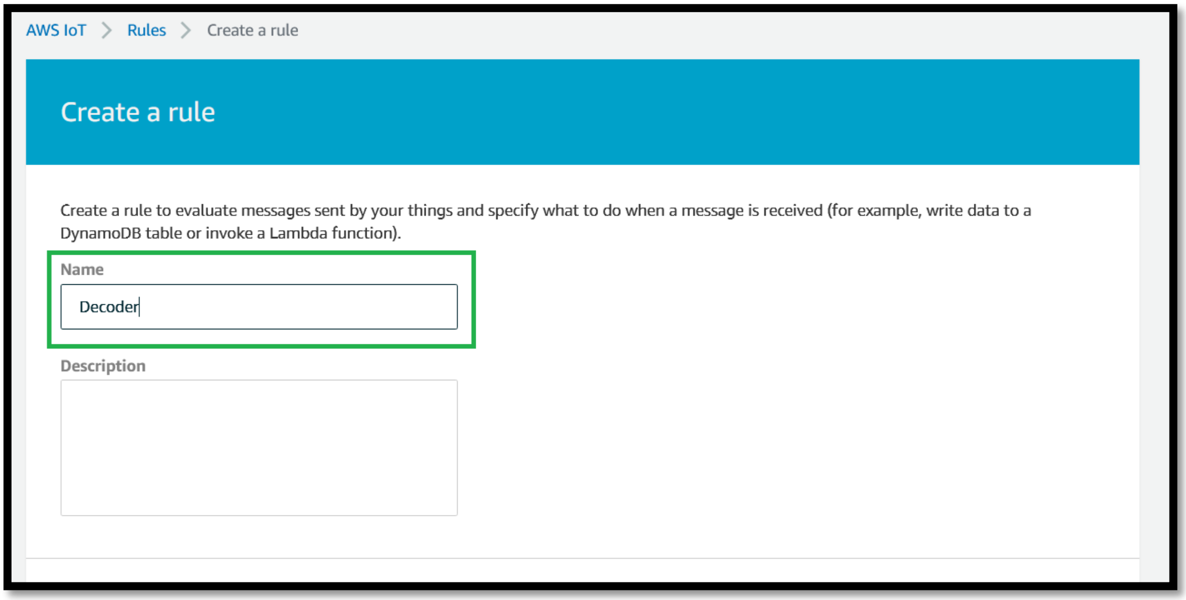


Figure 40: Creating a new Rule

Enter "Decoder" for the rule name as shown in Figure 41.



Create a rule

Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).

Name: Decoder

Description

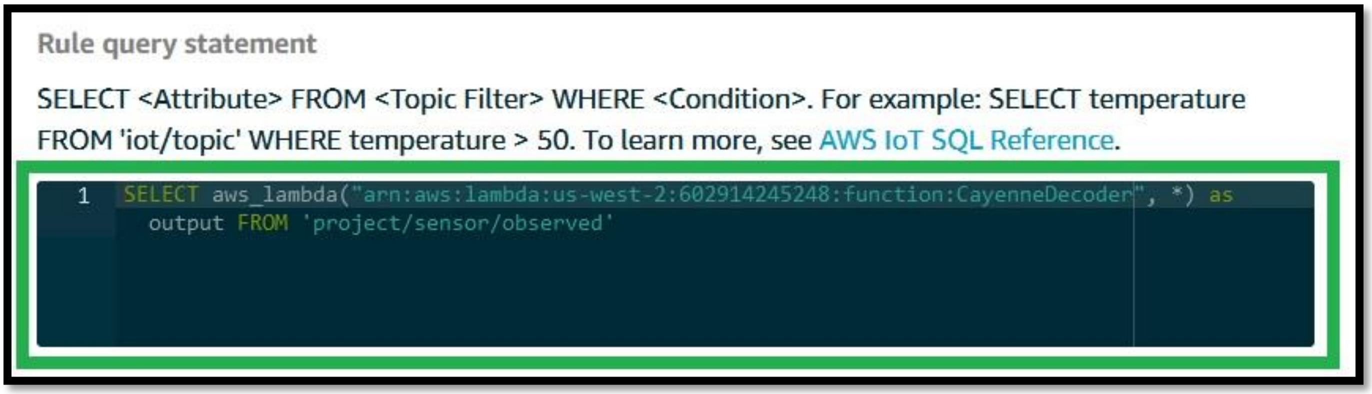
Figure 41: Creating the Decoder Rule

Scroll down to the query statement of the Rule, as shown in Figure 42. 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 4.4.3 for details of the Root MQTT Topic, this is enclosed within apostrophes.

Refer to Figure 42 for the expected formatting and appearance.



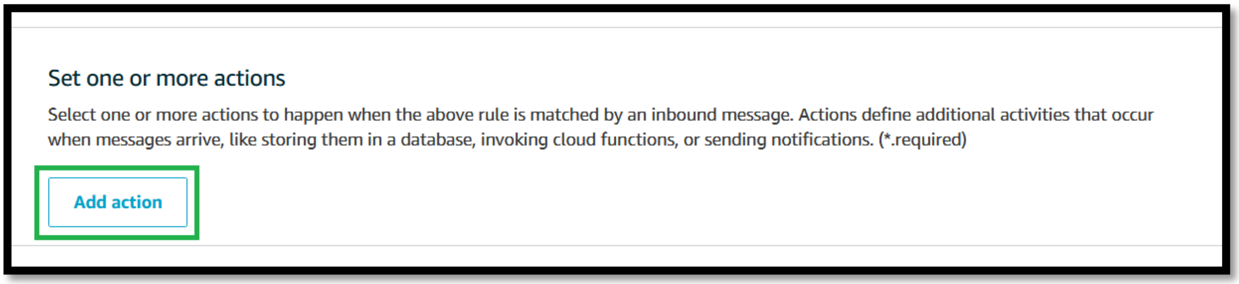
Rule query statement

SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see AWS IoT SQL Reference.

```
SELECT aws_lambda("arn:aws:lambda:us-west-2:602914245248:function:CayenneDecoder", *) as
  output FROM 'project/sensor/observed'
```

Figure 42: Rule query statement

You must add an Action to publish the output to an MQTT topic. As shown in Figure 43, click **Add Action** to add the publish action.

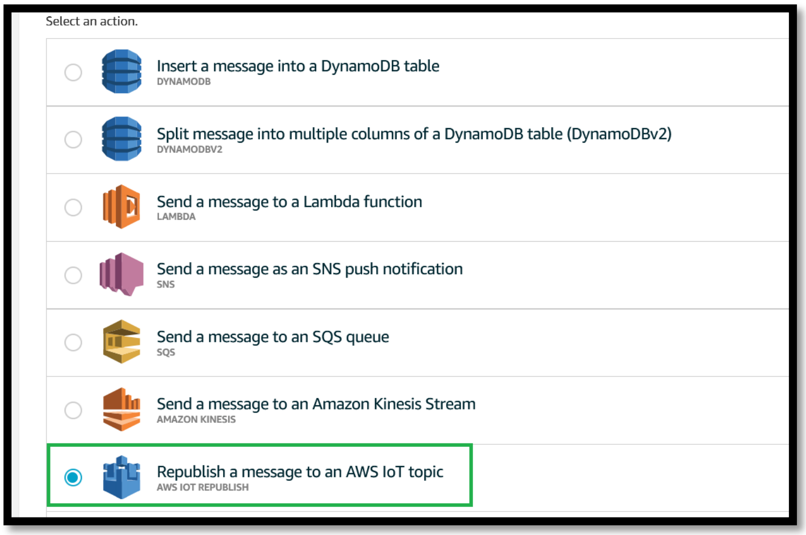


Set one or more actions

Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. (*.required)

Figure 43: Adding the publish Action

Select **Republish a message to an AWS IoT topic** as shown in Figure 44.



Select an action.

○ Insert a message into a DynamoDB table
DYNAMODB

○ Split message into multiple columns of a DynamoDB table (DynamoDBv2)
DYNAMODBV2

○ Send a message to a Lambda function
LAMBDA

○ Send a message as an SNS push notification
SNS

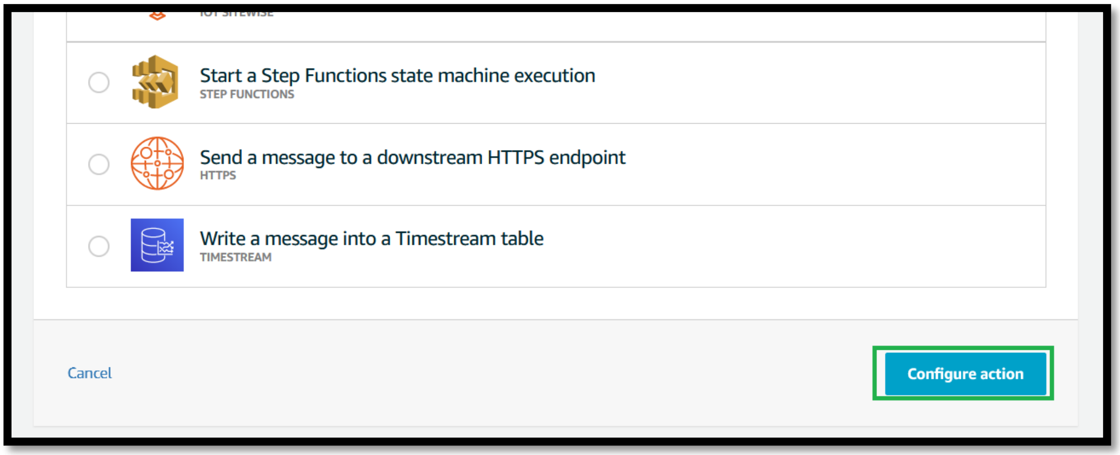
○ Send a message to an SQS queue
SQS

○ Send a message to an Amazon Kinesis Stream
AMAZON KINESIS

● Republish a message to an AWS IoT topic
AWS IOT REPUBLISH

Figure 44: Selecting the Republish Action

Scroll down and click **Configure action** as shown in Figure 45.



IOT SITEWISE

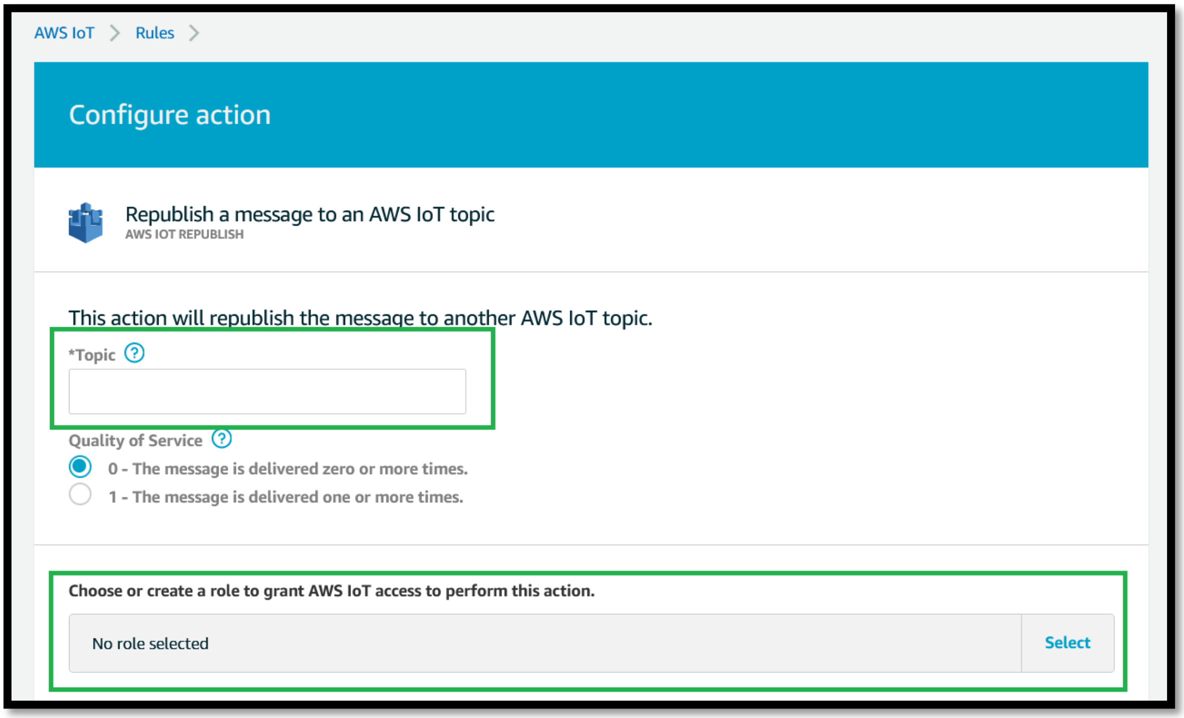
○ Start a Step Functions state machine execution
STEP FUNCTIONS

○ Send a message to a downstream HTTPS endpoint
HTTPS

○ Write a message into a Timestream table
TIMESTREAM

Figure 45: Configuring the republish Action

Figure 46 shows the Configure Action page. Set *Topic* to “Decoded”. Create a new Role to allow the publication by clicking **Select** and entering DecodedRole for the name of the Role.



AWS IoT > Rules >

Configure action

Republish a message to an AWS IoT topic
AWS IOT REPUBLISH

This action will republish the message to another AWS IoT topic.

*Topic

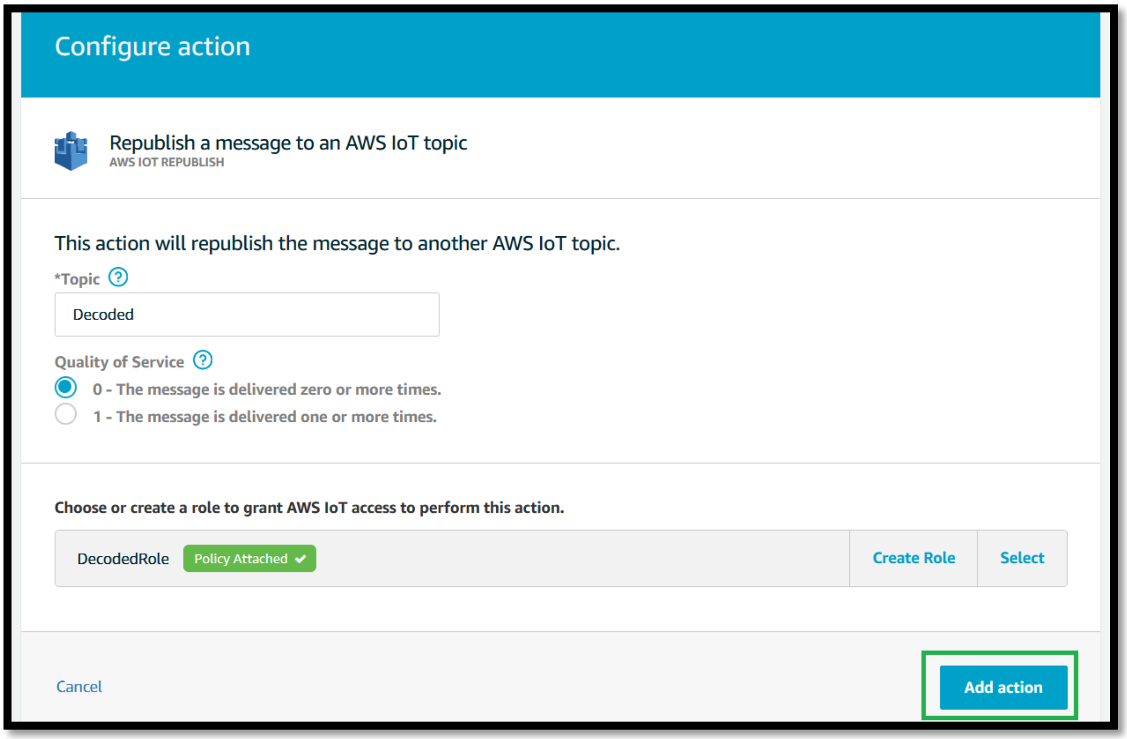
Quality of Service
● 0 - The message is delivered zero or more times.
○ 1 - The message is delivered one or more times.

Choose or create a role to grant AWS IoT access to perform this action.

No role selected | Select

Figure 46: Setting the republish topic

Click **Add action** as shown in Figure 47 to complete adding the Action.



Configure action

Republish a message to an AWS IoT topic
AWS IOT REPUBLISH

This action will republish the message to another AWS IoT topic.

*Topic
Decoded

Quality of Service
● 0 - The message is delivered zero or more times.
○ 1 - The message is delivered one or more times.

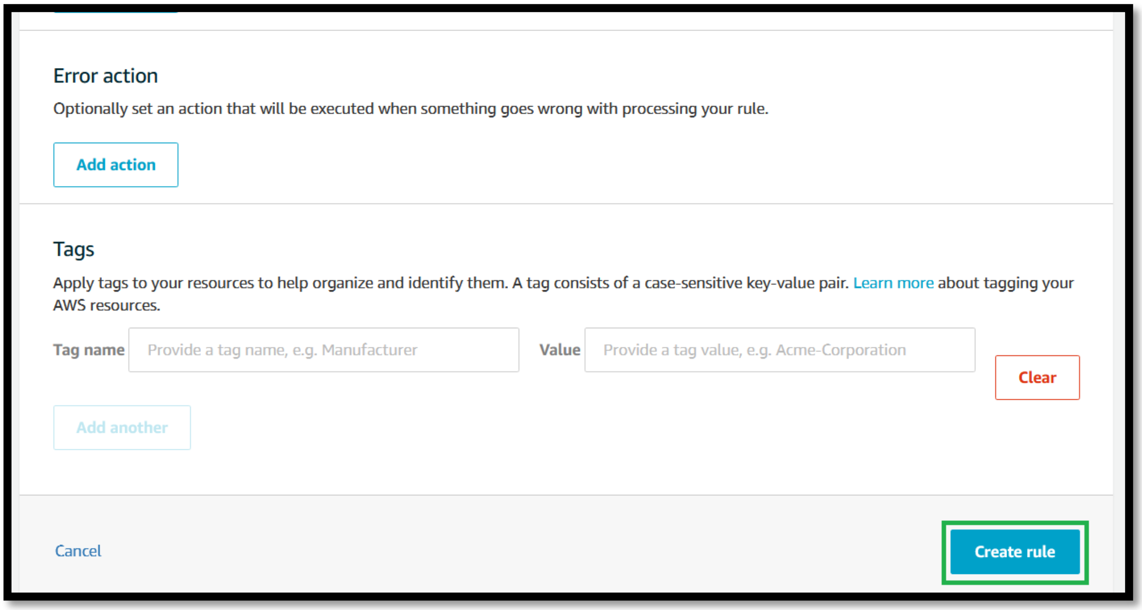
Choose or create a role to grant AWS IoT access to perform this action.

DecodedRole Policy Attached ✓ | Create Role | Select

Cancel | Add action

Figure 47: Adding the republish Action

Create the Decoded Rule by clicking **Create rule** as shown in Figure 48.



Error action

Optionally set an action that will be executed when something goes wrong with processing your rule.

Tags

Apply tags to your resources to help organize and identify them. A tag consists of a case-sensitive key-value pair. Learn more about tagging your AWS resources.

Tag name Provide a tag name, e.g. Manufacturer

Value Provide a tag value, e.g. Acme-Corporation

Figure 48: Creating the Decoded Rule

From the Rules screen, click **Add Action** again, and then select *Send a message to a Lambda function*, as shown in Figure 49.

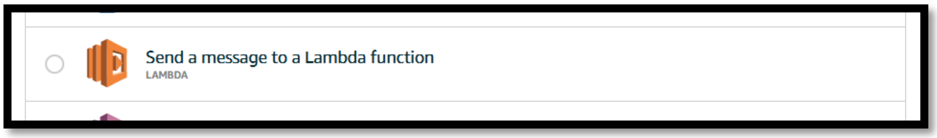
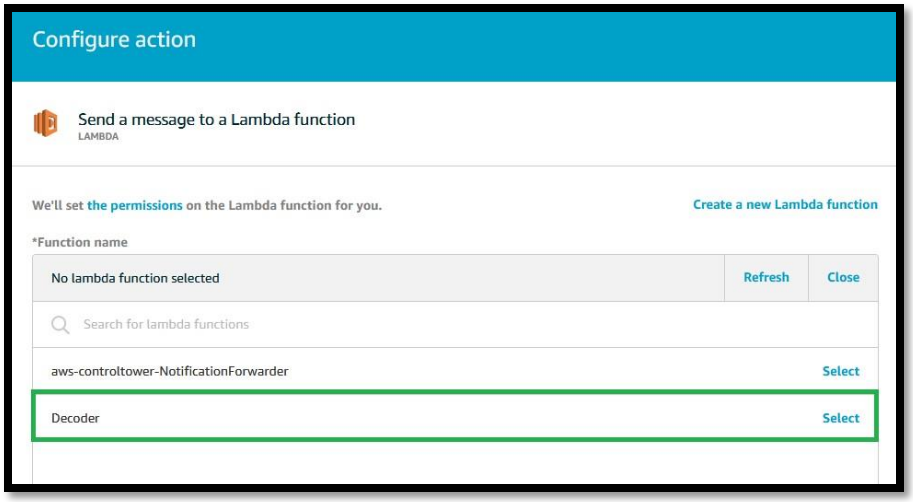


Figure 49: Adding the Send a message to a Lambda function

On the *Configure action* screen, the Decoder lambda should be selected from the Function Name drop down list, as shown in Figure 50.



Configure action

Send a message to a Lambda function
LAMBDA

We'll set the permissions on the Lambda function for you.

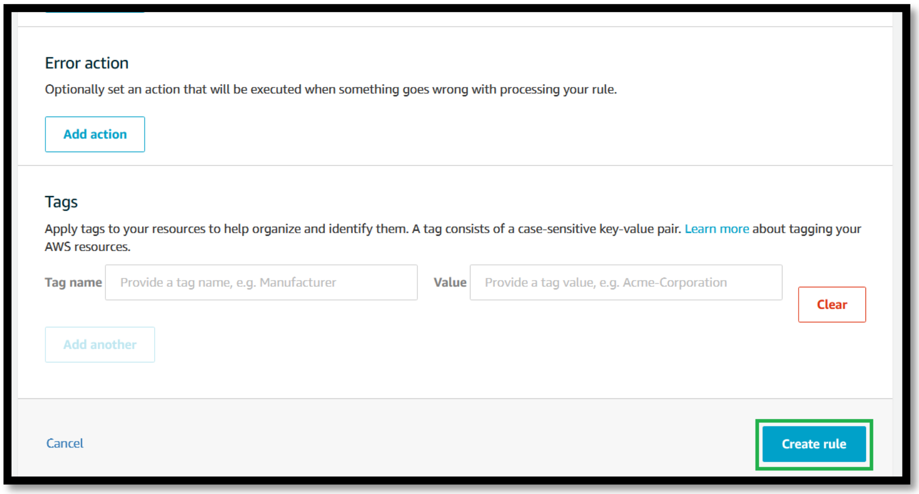
*Function name

No lambda function selected

aws-controltower-NotificationForwarder	Select
Decoder	Select

Figure 50: Selecting the Decoder Lambda

Click **Add action,** then click **Create rule** as shown in Figure 51.



Error action

Optionally set an action that will be executed when something goes wrong with processing your rule.

Tags

Apply tags to your resources to help organize and identify them. A tag consists of a case-sensitive key-value pair. Learn more about tagging your AWS resources.

Tag name
Provide a tag name, e.g. Manufacturer
Value
Provide a tag value, e.g. Acme-Corporation
Clear
Add another
Cancel
Create rule

Figure 51: Creating the Decoder Rule

6.1.3 Granting Lambda code access to the IoT Core

In the *Rules* menu, click the ellipsis to the right of the Decoder rule and click **Enable**.

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.

6.1.4 Creating the Extractor Rule and Extracted Database

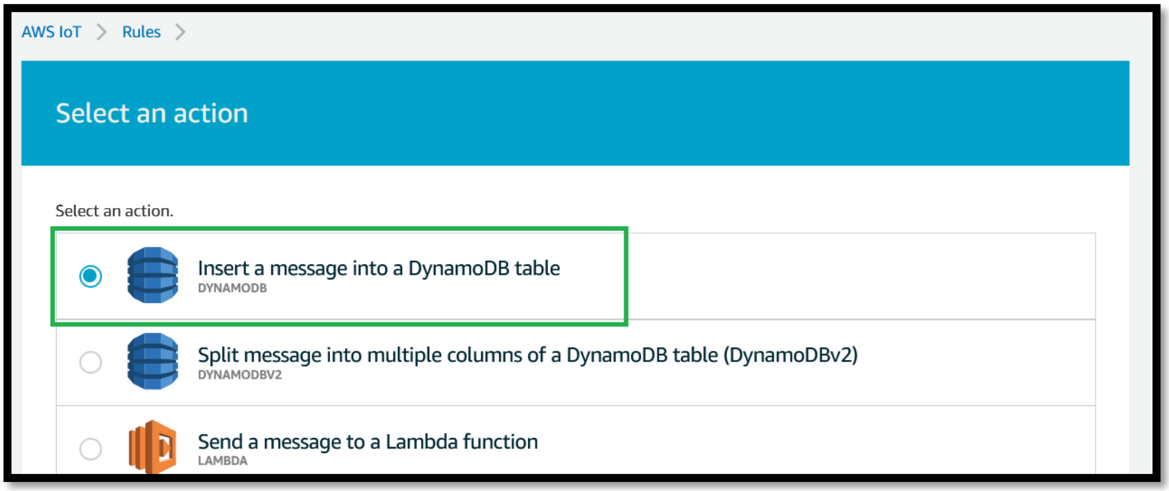
You must create a further Rule called “Extractor” as described in section 6.1.2. Set the query for this Rule as follows:

```
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 Actions, create a second republish 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 **Insert a message into a DynamoDB table**, then click **Configure action** as shown in Figure 52.



Select an action

Select an action.

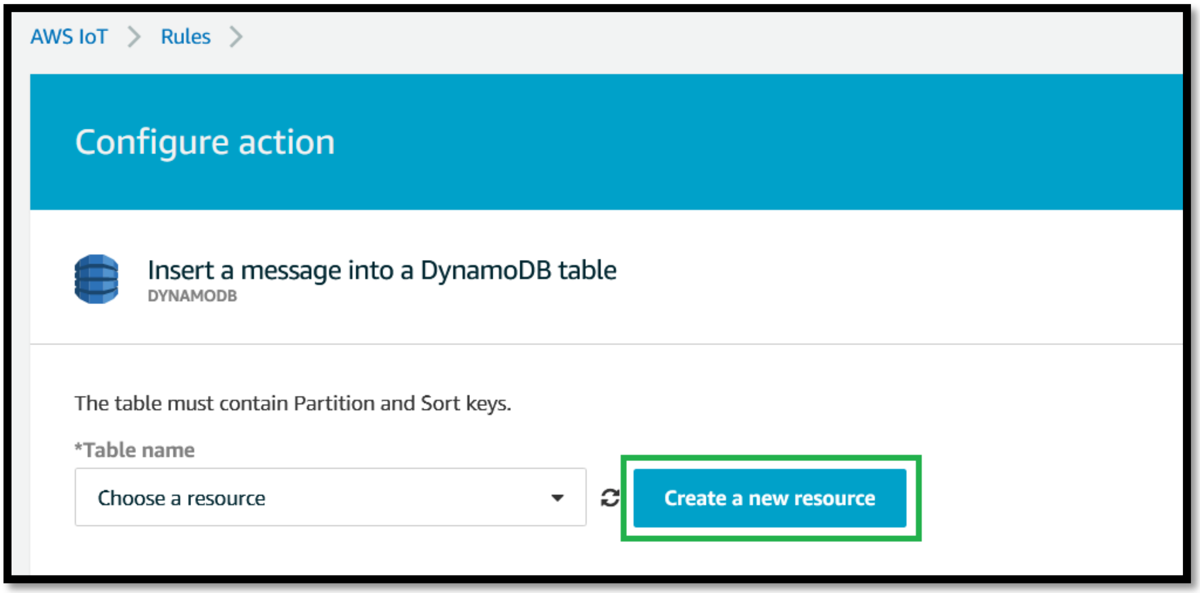
● Insert a message into a DynamoDB table
DYNAMODB

○ Split message into multiple columns of a DynamoDB table (DynamoDBv2)
DYNAMODBV2

○ Send a message to a Lambda function
LAMBDA

Figure 52: Adding the insert message into a DynamoDB table action

Unless a table is already available, you'll need to create one. Click **Create new resource** as shown in Figure 53.



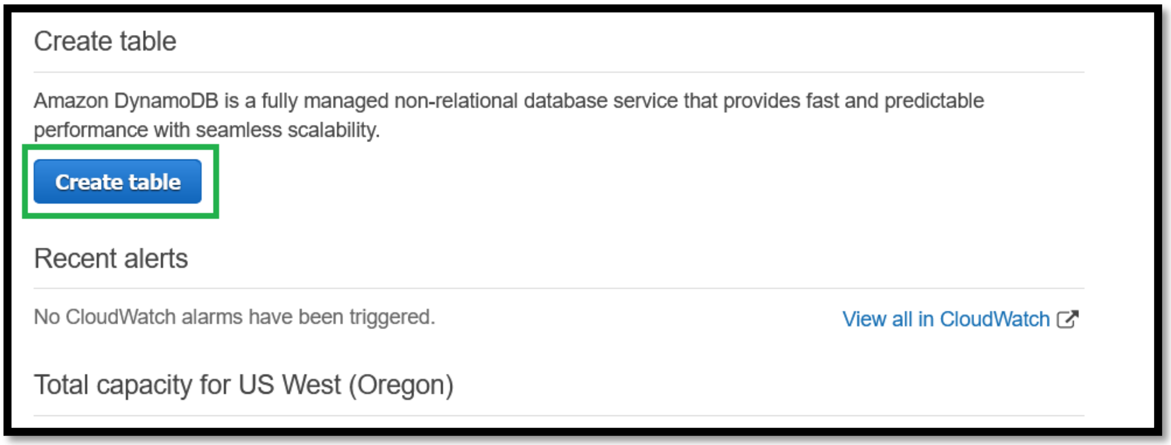
The table must contain Partition and Sort keys.

*Table name

Choose a resource

Figure 53: Configuring the database action

This opens the DynamoDB main page, as shown in Figure 54. Click **Create table** to create the database table where published data will be stored.



Create table

Amazon DynamoDB is a fully managed non-relational database service that provides fast and predictable performance with seamless scalability.

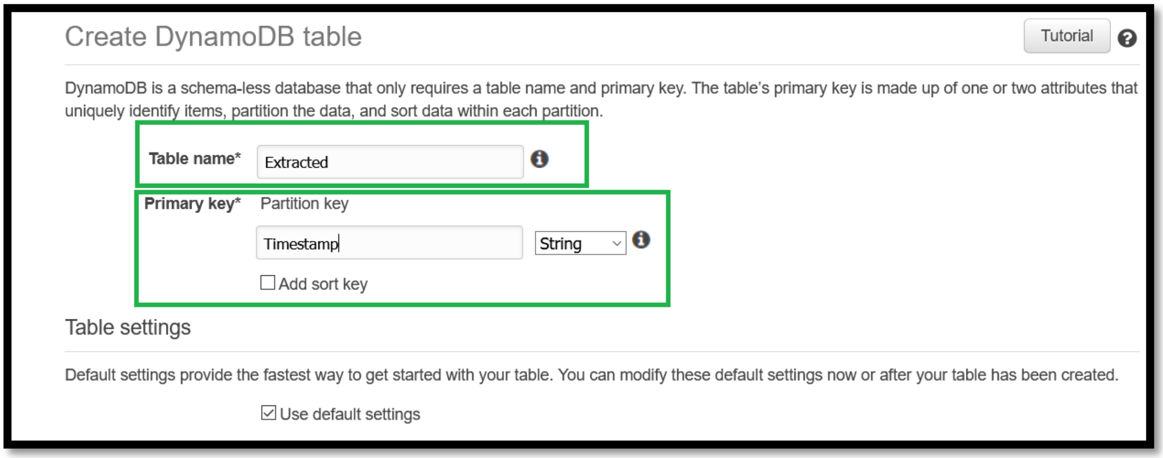
Recent alerts

No CloudWatch alarms have been triggered.

Total capacity for US West (Oregon)

Figure 54: DynamoDB main page

Set *Table Name* to "Extracted" *Primary Key* to "Timestamp" as shown in Figure 55.



Create DynamoDB table

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* Extracted

Primary key* Partition key

Timestamp | String

☐ Add sort key

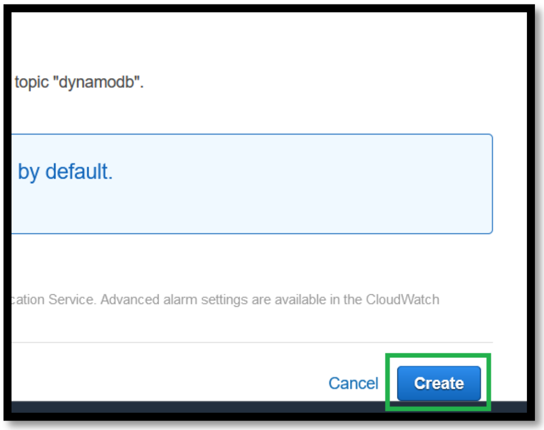
Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

☑ Use default settings

Figure 55: Adding the database table

Scroll down and click **Create** to finalise the table.



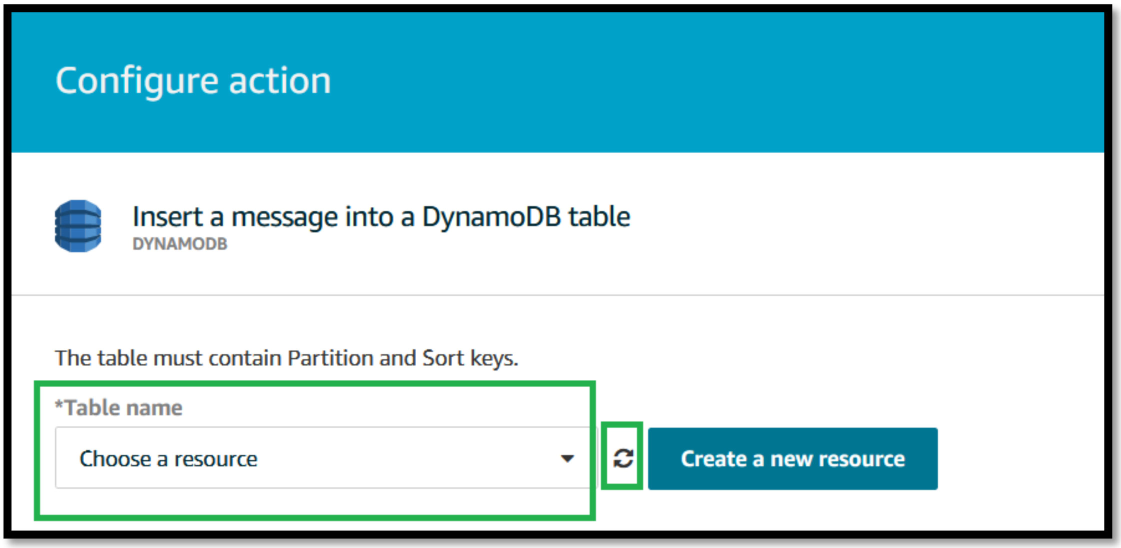
topic "dynamodb".

by default.

ation Service. Advanced alarm settings are available in the CloudWatch

Figure 56: Finalising creation of the database table

Return to the “Configure action” page. Click **Refresh** to update the list of available tables. Select *Extracted* as shown in Figure 57.



Configure action

Insert a message into a DynamoDB table
DYNAMODB

The table must contain Partition and Sort keys.

*Table name

Choose a resource

Figure 57: Selecting the Extracted database

Set *Partition Key* as follows:

```
${output.timestamp}
```

This uses the message timestamp as the primary key for the table, ensuring each entry is unique.

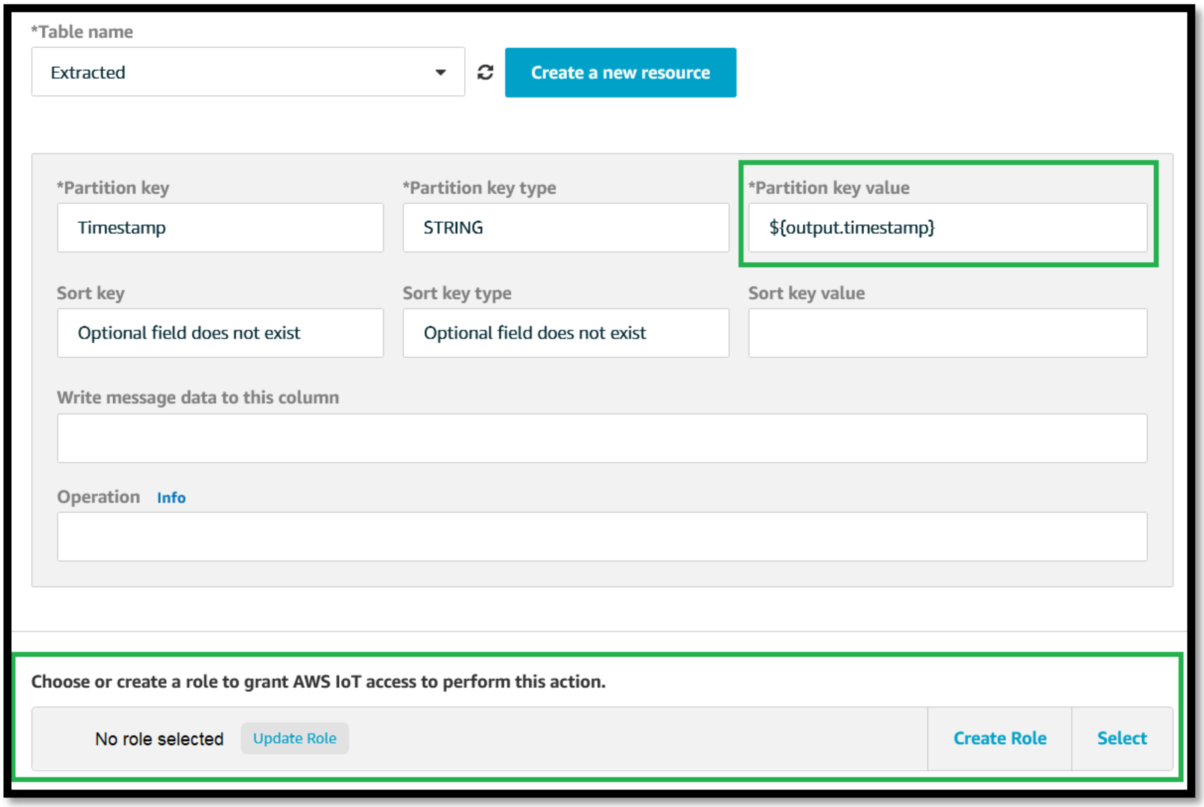


*Table name: Extracted

*Partition key	*Partition key type	*Partition key value
Timestamp	STRING	${output.timestamp}

Sort key	Sort key type	Sort key value
Optional field does not exist	Optional field does not exist	

Write message data to this column

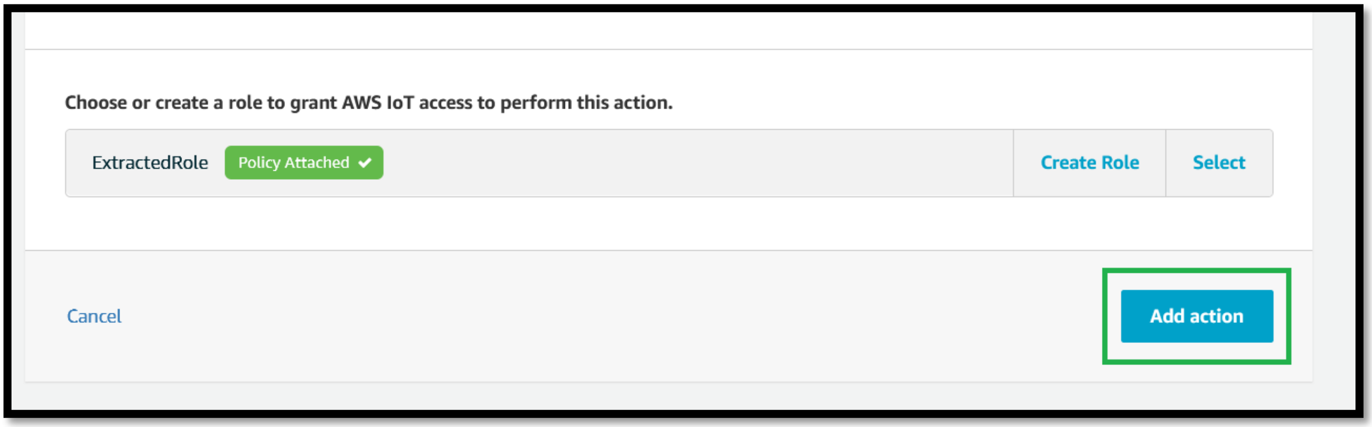
Operation Info

Choose or create a role to grant AWS IoT access to perform this action.

No role selected

Figure 58: Configuring the database action

Create a Role to allow updates to be made to the database. Select **Create role** and set *Role Name* to “ExtractedRole” as shown in Figure 59. Click **Add action** to finalize the Action.

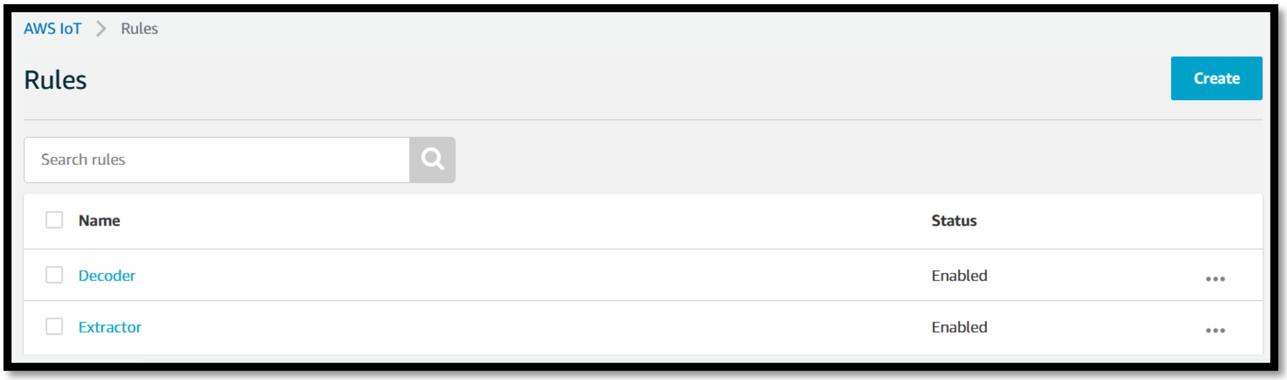

Choose or create a role to grant AWS IoT access to perform this action.

ExtractedRole

Figure 59: Adding the database Role

6.1.5 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 60.



□ Name	Status	
□ Decoder	Enabled	•••
□ Extractor	Enabled	•••

Figure 60: Enabling the Decoder and Extractor Rules

Click the ellipsis to the right of each Rule and select *Enabled*.

6.1.6 Testing the application

From the AWS IoT page, click **Test** in the left menu. This opens the MQTT Client as shown in Figure 61.

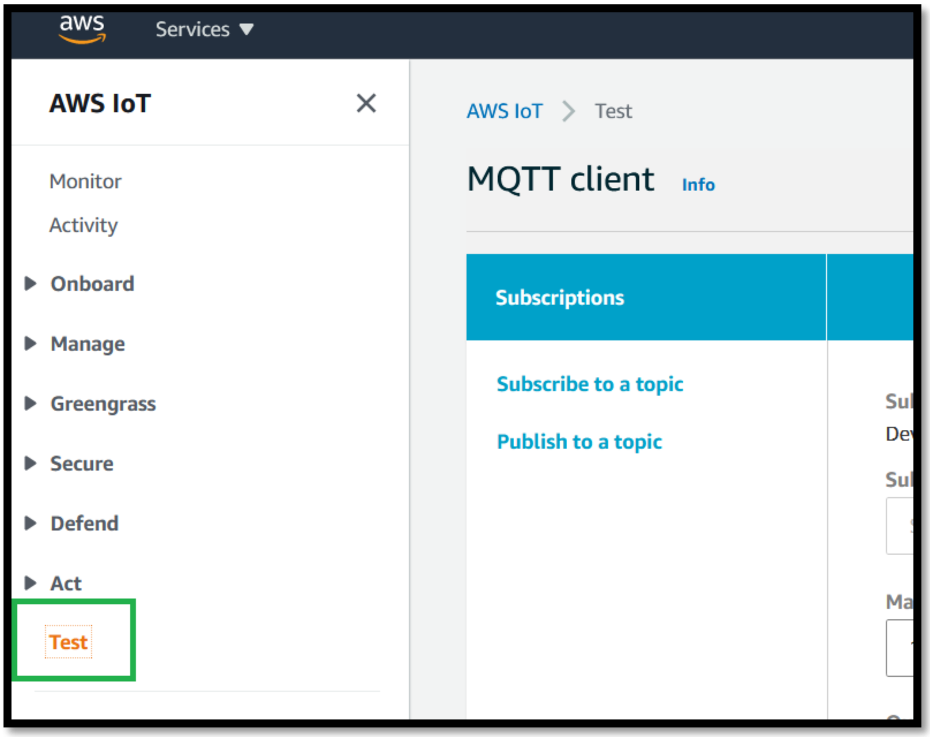
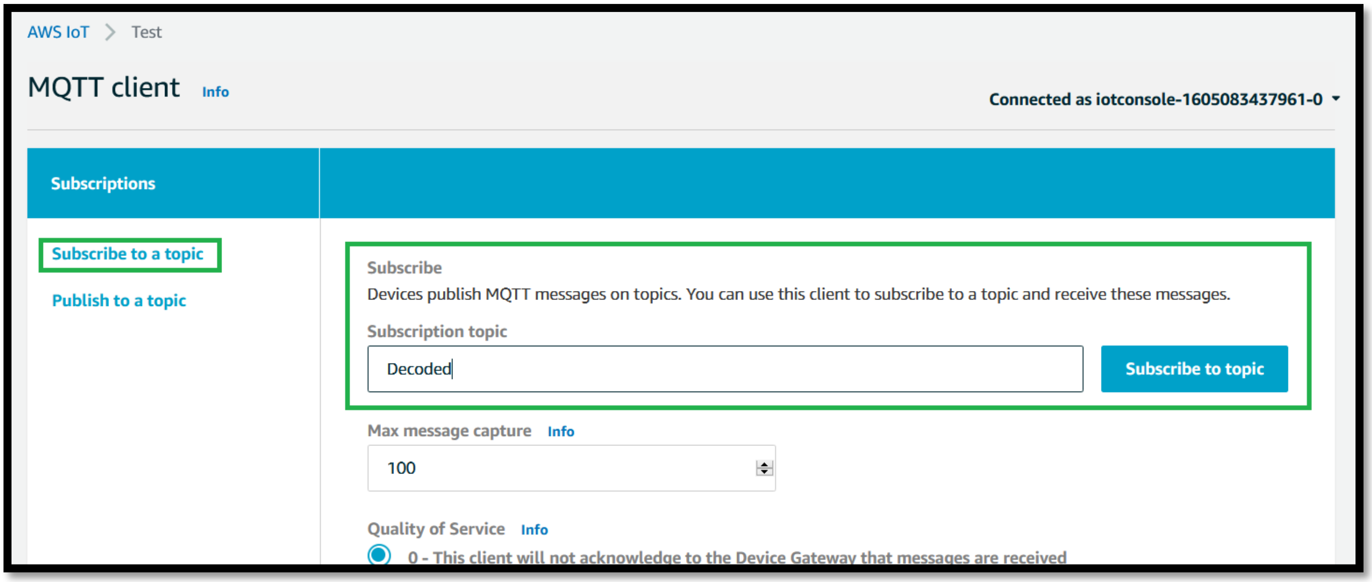


Figure 61: Opening the IoT Core MQTT Client

Click **Subscribe to a topic**, then set *Subscription Topic* to "Decoded" and click **Subscribe to topic** as shown in Figure 62.



Subscribe

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

Subscription topic: Decoded

Max message capture: 100

Quality of Service

● 0 - This client will not acknowledge to the Device Gateway that messages are received

Figure 62: Subscribing to the Decoded topic

Repeat the process for the "Extracted" topic.

Available subscriptions display to the left of the MQTT Client page as shown in Figure 63.

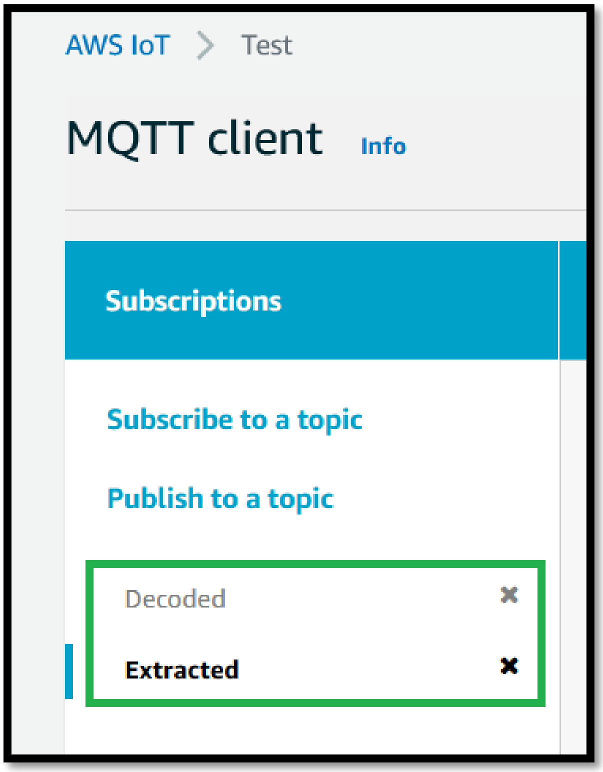
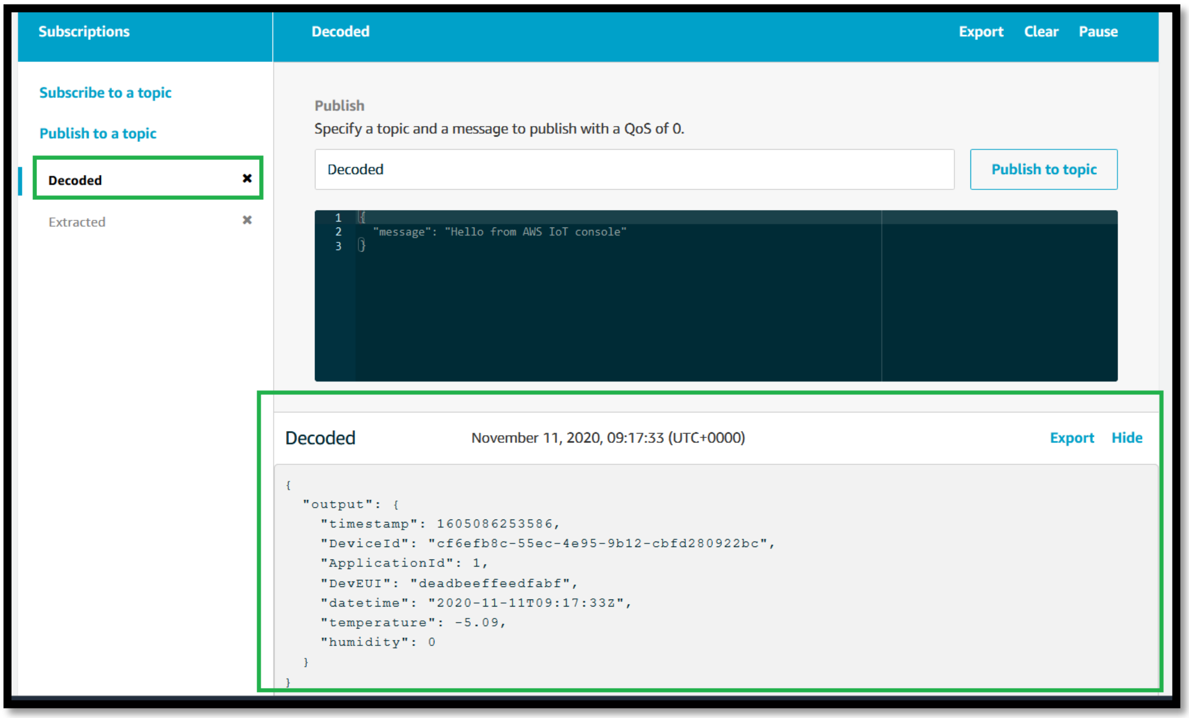


Figure 63: Available topic subscriptions

Click a subscription and scroll down to see the incoming message data, as shown in Figure 64.



Specify a topic and a message to publish with a QoS of 0.

Decoded

```
{
  "message": "Hello from AWS IoT console"
}
```

Decoded November 11, 2020, 09:17:33 (UTC+0000)

```
{
  "output": {
    "timestamp": 1605086253586,
    "DeviceId": "cf6efb8c-55ec-4e95-9b12-cbfd280922bc",
    "ApplicationId": 1,
    "DevEUI": "deadbeeffeedfabf",
    "datetime": "2020-11-11T09:17:33Z",
    "temperature": -5.09,
    "humidity": 0
  }
}
```

Figure 64: Observing incoming Decoded topic messages

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 65.

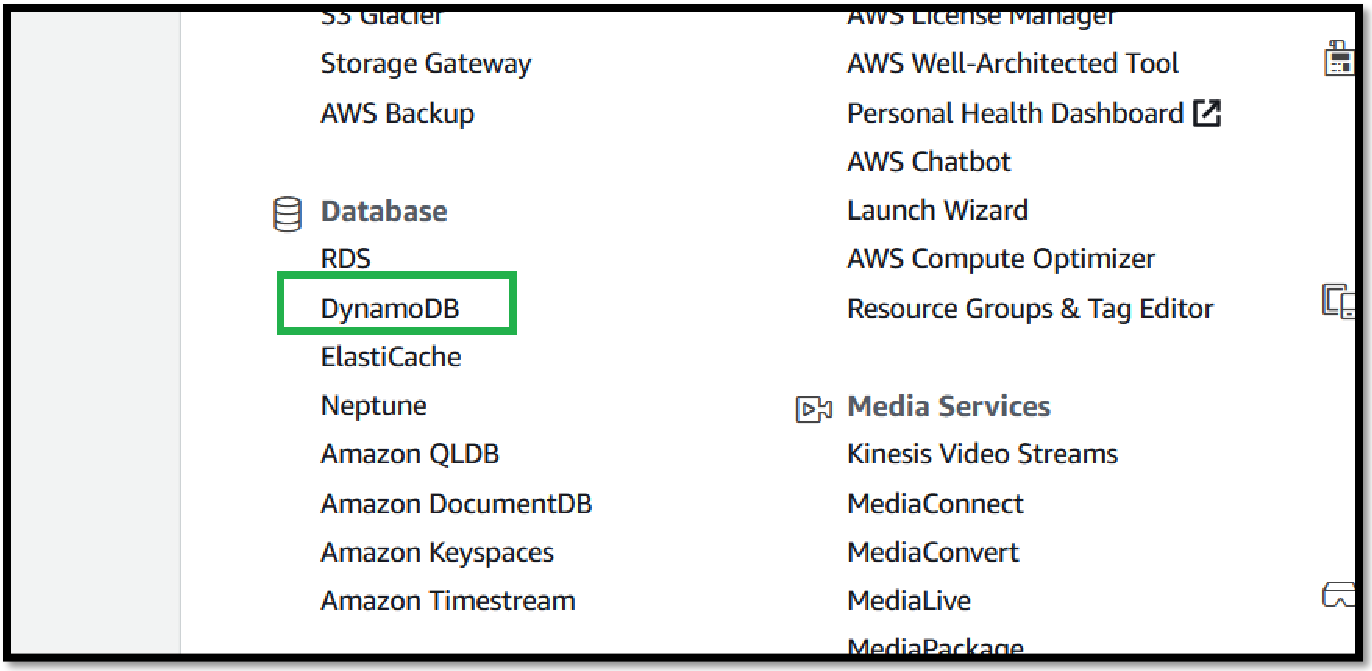
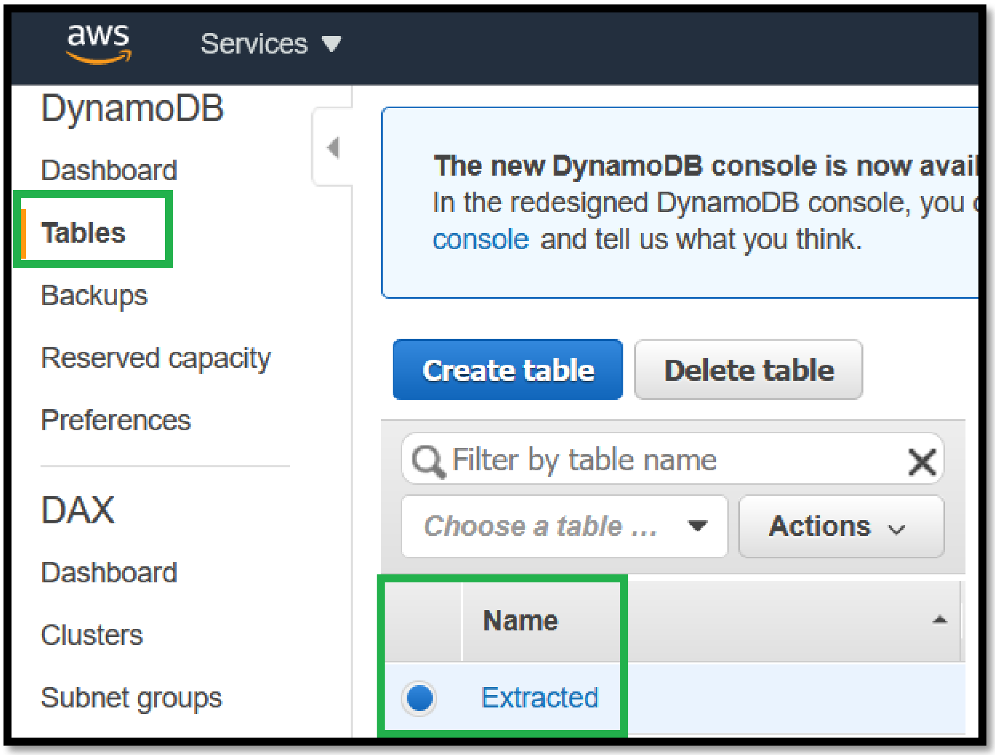


Figure 65: Opening the DynamoDB service from the AWS Management Console

Click **Tables** in the submenu on the left, then select *Extracted* as shown in Figure 66.

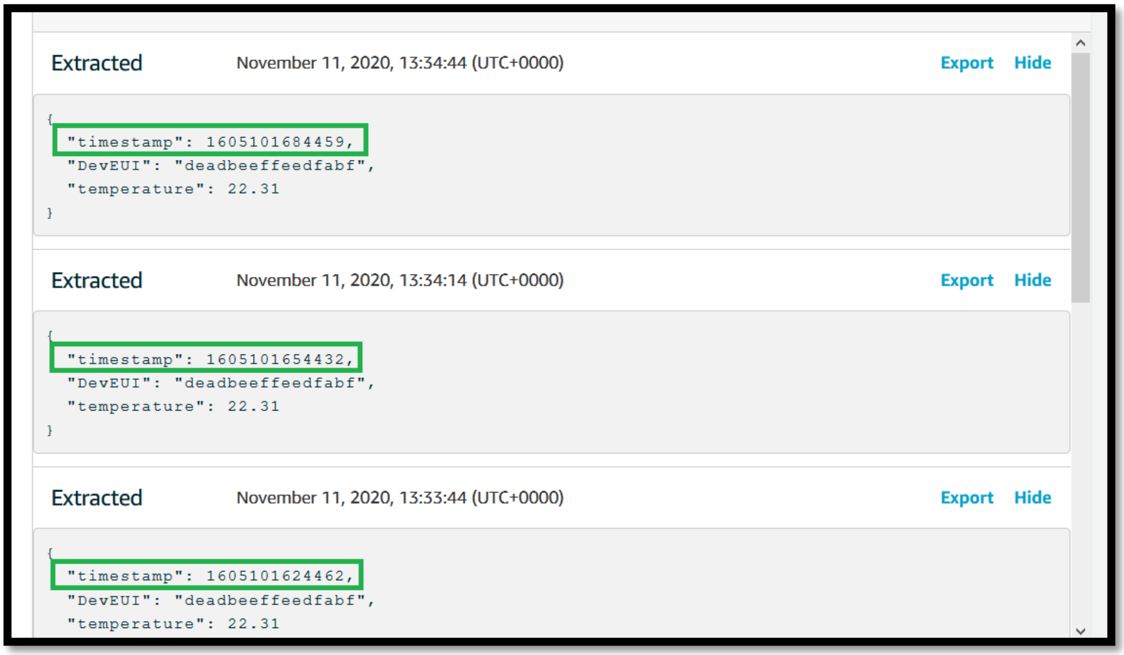


The new DynamoDB console is now avail
In the redesigned DynamoDB console, you c
console and tell us what you think.

	Name
●	Extracted

Figure 66: 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 67.



Extracted November 11, 2020, 13:34:44 (UTC+0000) Export Hide

```
{
  "timestamp": 1605101684459,
  "DevEUI": "deadbeeffeedfabf",
  "temperature": 22.31
}
```

Extracted November 11, 2020, 13:34:14 (UTC+0000) Export Hide

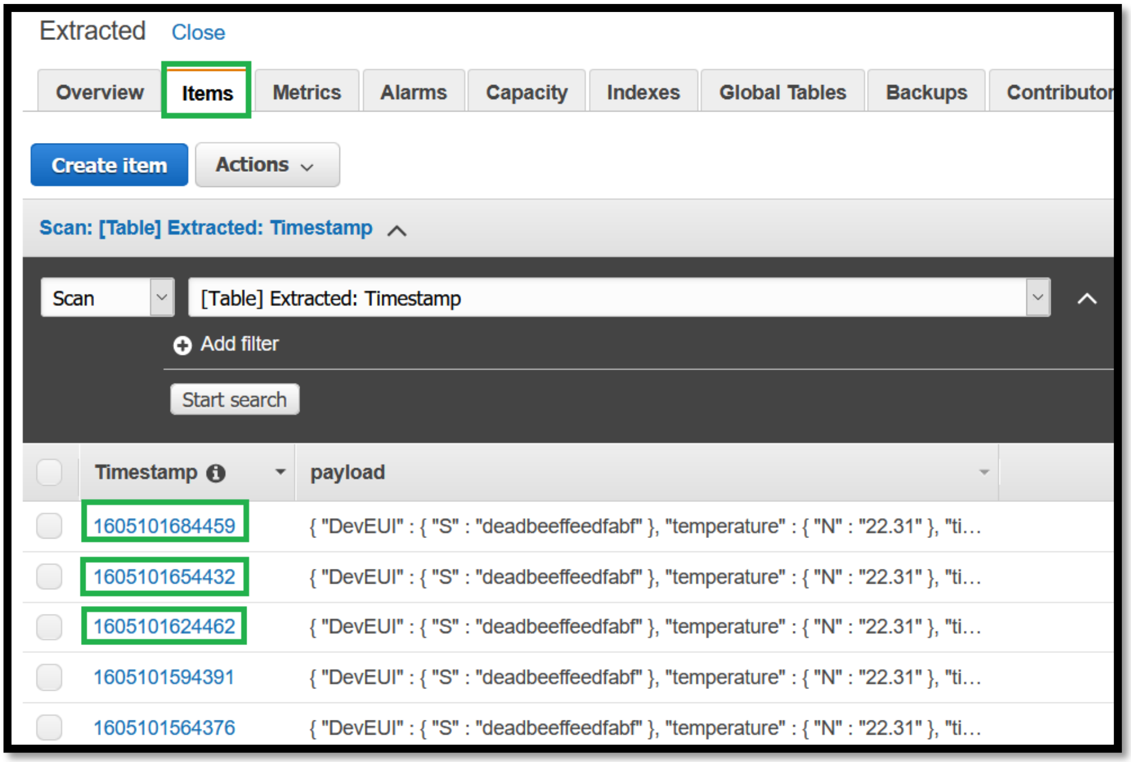
```
{
  "timestamp": 1605101654432,
  "DevEUI": "deadbeeffeedfabf",
  "temperature": 22.31
}
```

Extracted November 11, 2020, 13:33:44 (UTC+0000) Export Hide

```
{
  "timestamp": 1605101624462,
  "DevEUI": "deadbeeffeedfabf",
  "temperature": 22.31
```

Figure 67: 'Decoded' topic messages

From the DynamoDB page, click **Items** to view the "Extracted" table contents as shown in Figure 68. There should be parity between the content of the AWS MQTT Client and the table.



Scan: [Table] Extracted: Timestamp

Scan [Table] Extracted: Timestamp

	Timestamp	payload
□	1605101684459	{ "DevEUI" : { "S" : "deadbeeffeedfabf" }, "temperature" : { "N" : "22.31" }, "ti…
□	1605101654432	{ "DevEUI" : { "S" : "deadbeeffeedfabf" }, "temperature" : { "N" : "22.31" }, "ti…
□	1605101624462	{ "DevEUI" : { "S" : "deadbeeffeedfabf" }, "temperature" : { "N" : "22.31" }, "ti…
□	1605101594391	{ "DevEUI" : { "S" : "deadbeeffeedfabf" }, "temperature" : { "N" : "22.31" }, "ti…
□	1605101564376	{ "DevEUI" : { "S" : "deadbeeffeedfabf" }, "temperature" : { "N" : "22.31" }, "ti…

Figure 68: Viewing 'Extracted' topic messages being added to the 'Extracted' table

6.2 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 v10.x format for decoding the data packets is available from our GitHub page at reference [W].

Figure 69 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.

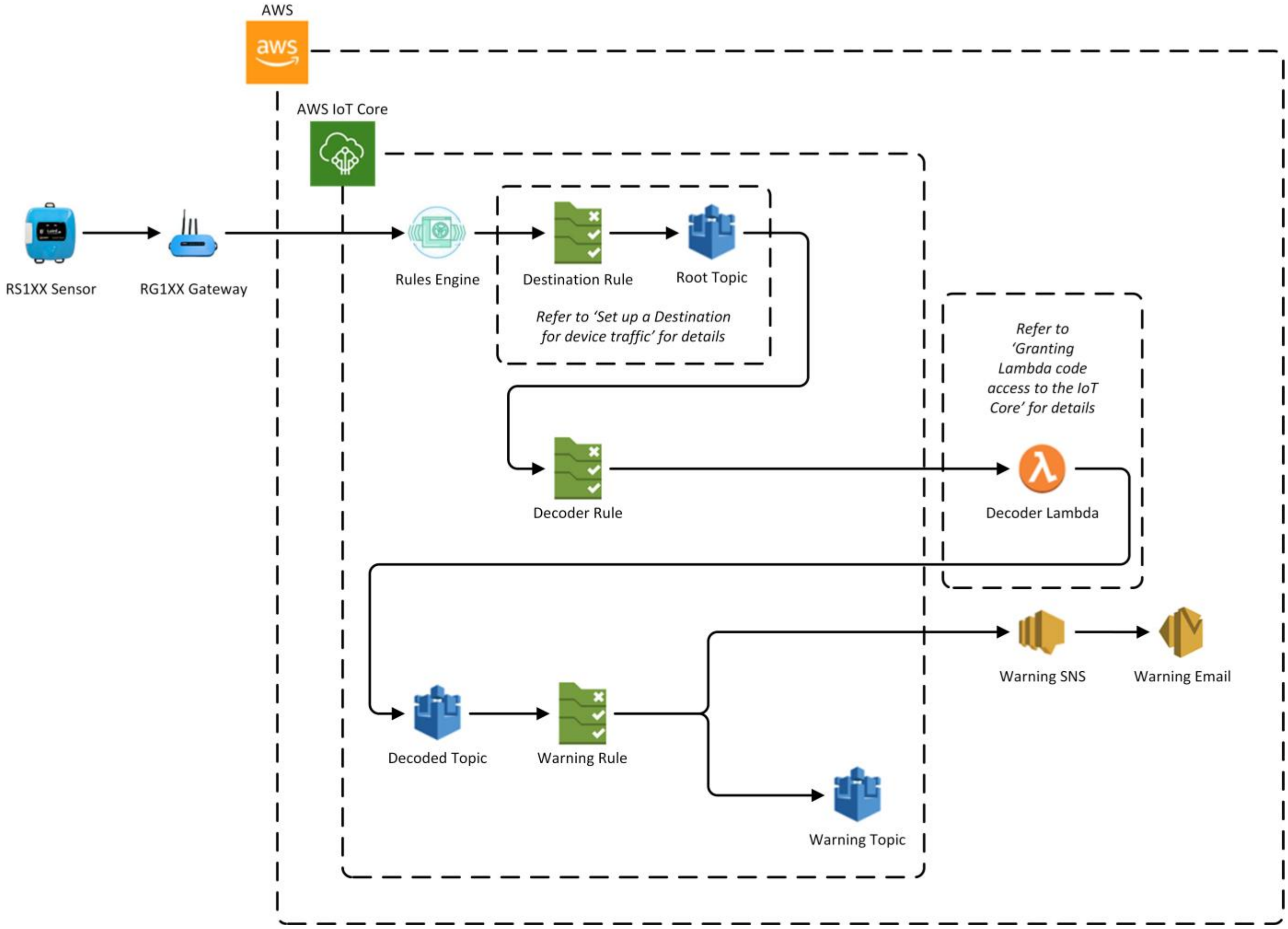


Figure 69: Cayenne Packet Format application architecture

6.2.1 Creating the Decoder Lambda function

Follow the steps in section 6.1.1 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.

6.2.2 Creating the Decoder Rule and Decoded topic

Repeat the steps in section 6.1.2 and 6.1.3.

6.2.3 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 70.

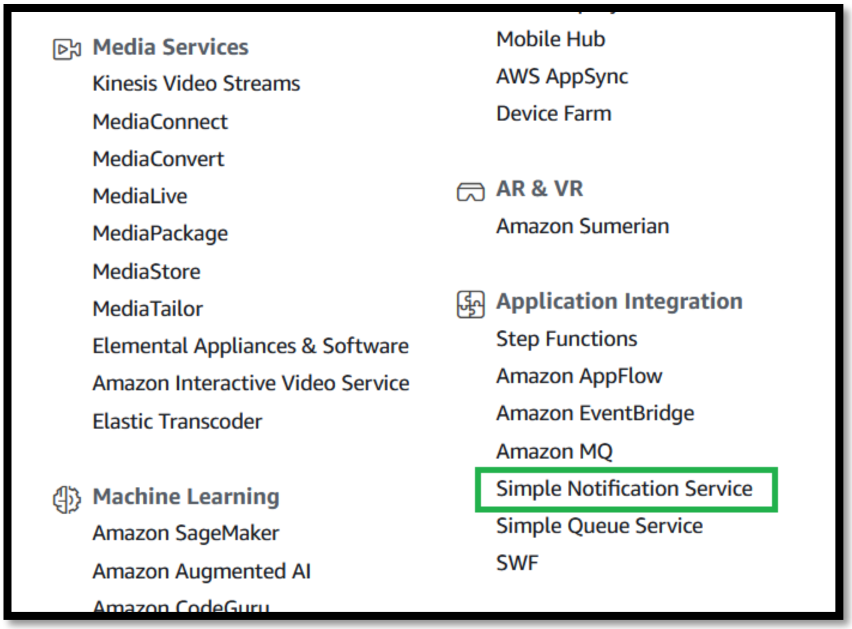
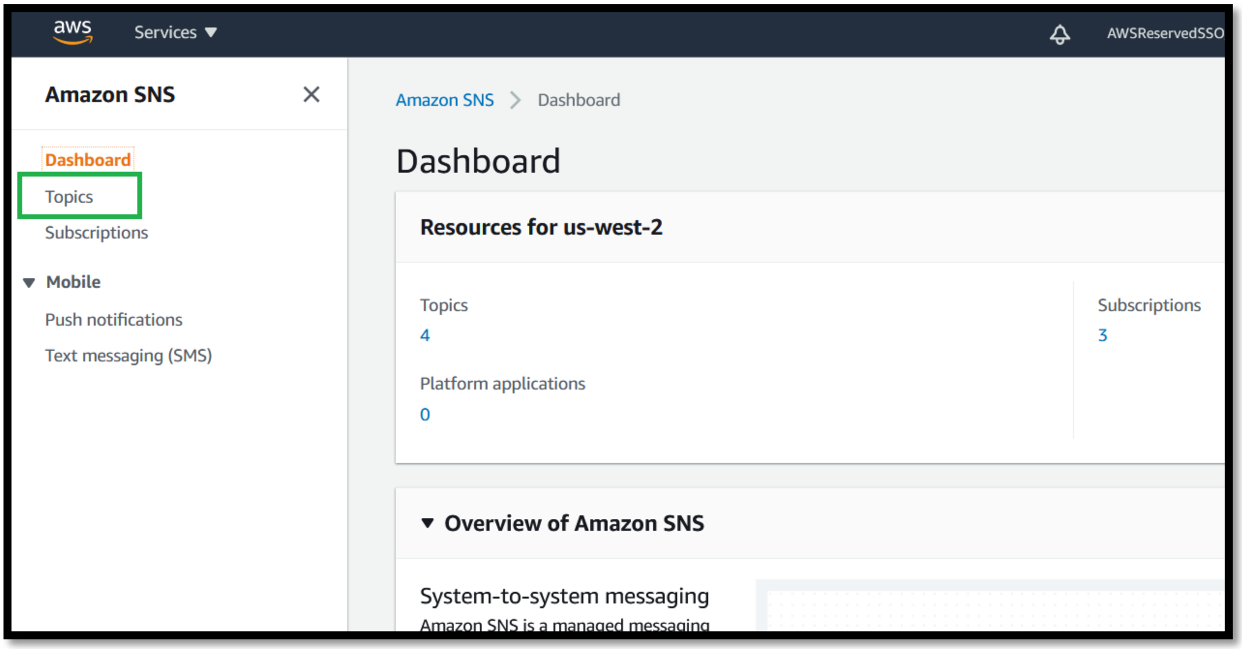


Figure 70: Opening the SNS page

This opens the Simple Notification Service page as shown in Figure 71. In the left submenu click **Topics**.



Dashboard

Resources for us-west-2

Topics
4

Subscriptions
3

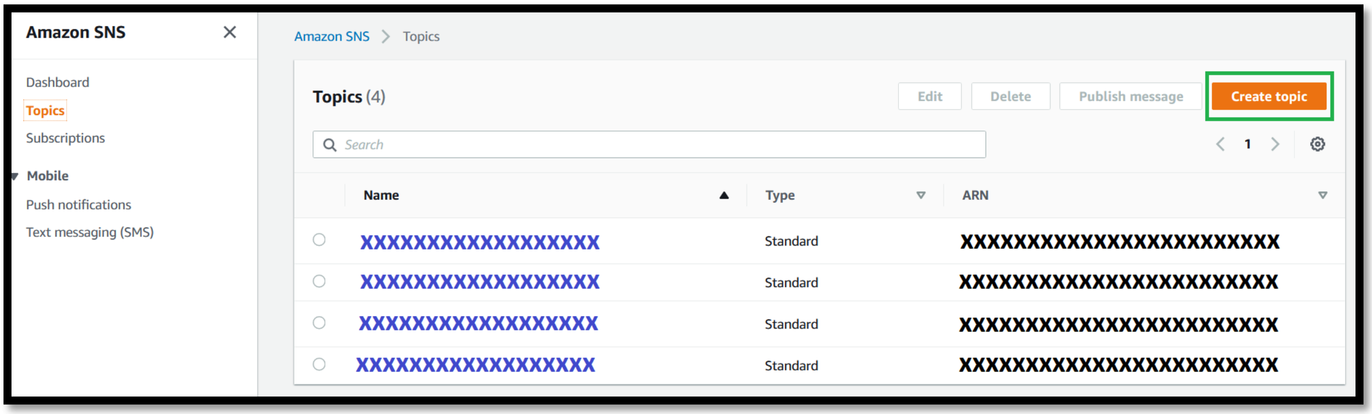
Platform applications
0

▼ Overview of Amazon SNS

System-to-system messaging
Amazon SNS is a managed messaging

Figure 71: SNS main page

Click **Create topic** as shown in Figure 72. This facilitates sending emails when appropriate messages are published.

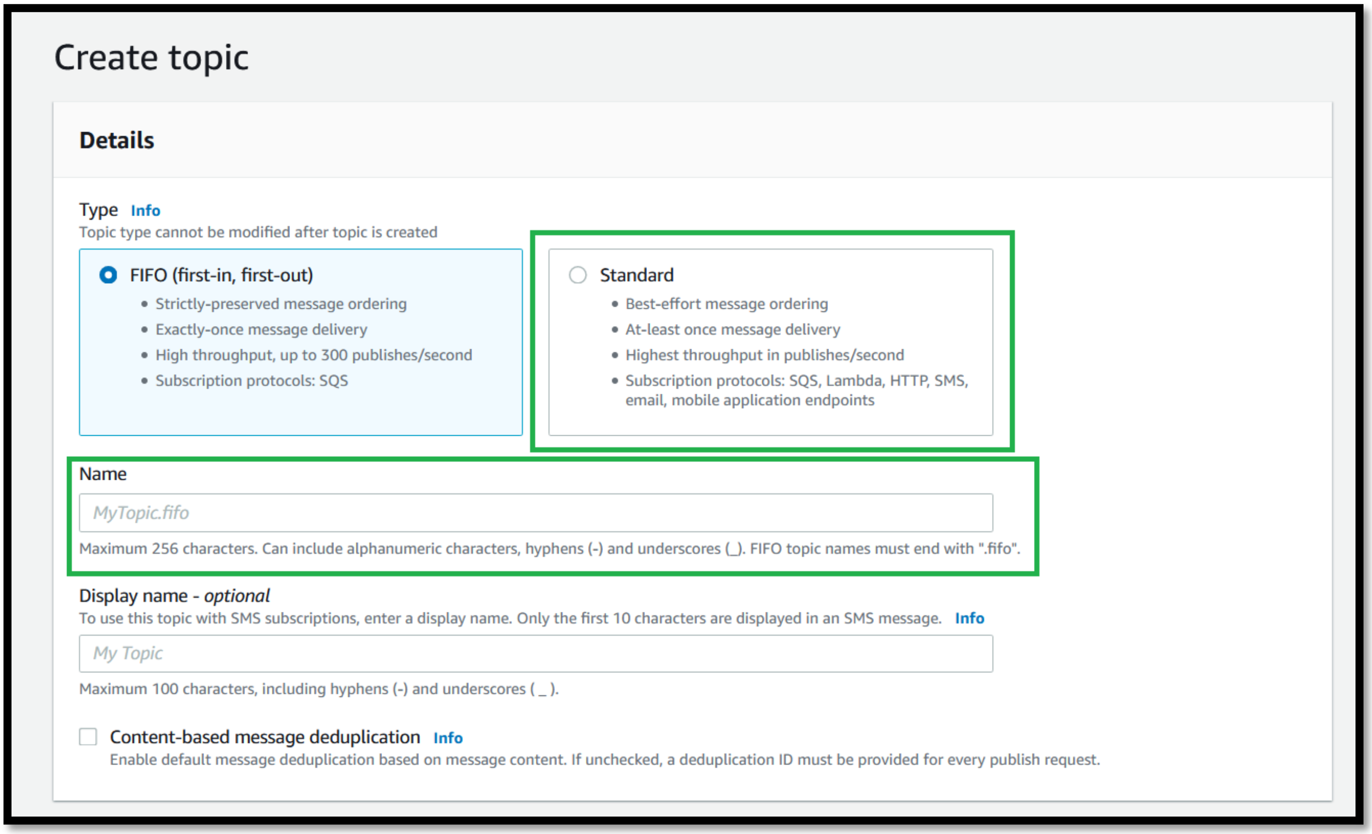


Topics (4)

	Name ▲	Type	ARN
○	XXXXXXXXXXXXXXXXXX	Standard	XXXXXXXXXXXXXXXXXXXXXXXX
○	XXXXXXXXXXXXXXXXXX	Standard	XXXXXXXXXXXXXXXXXXXXXXXX
○	XXXXXXXXXXXXXXXXXX	Standard	XXXXXXXXXXXXXXXXXXXXXXXX
○	XXXXXXXXXXXXXXXXXX	Standard	XXXXXXXXXXXXXXXXXXXXXXXX

Figure 72: SNS Topic list

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



Create topic

Details

Type
Topic type cannot be modified after topic is created

● FIFO (first-in, first-out)
- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

○ Standard
- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

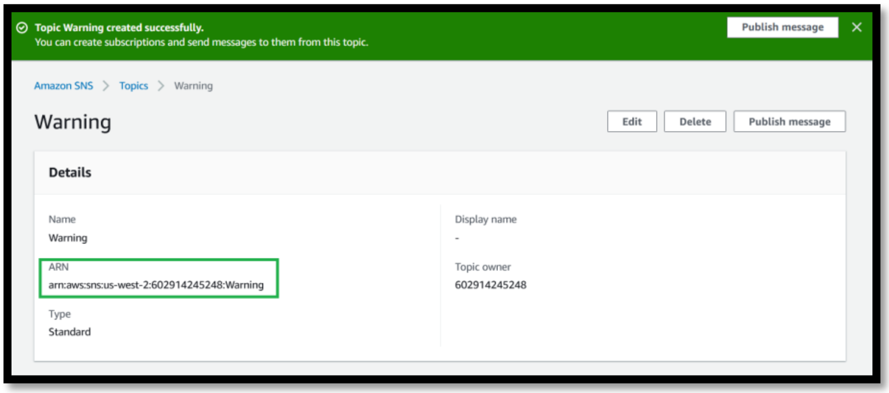
Name
MyTopic.fifo
Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_). FIFO topic names must end with ".fifo".

Display name - *optional*
To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.
My Topic
Maximum 100 characters, including hyphens (-) and underscores (_).

□ Content-based message deduplication
Enable default message deduplication based on message content. If unchecked, a deduplication ID must be provided for every publish request.

Figure 73: Creating an SNS topic

If successful, the new topic page displays as shown in Figure 74. Note the ARN of the topic for later use.



Topic Warning created successfully.
You can create subscriptions and send messages to them from this topic.

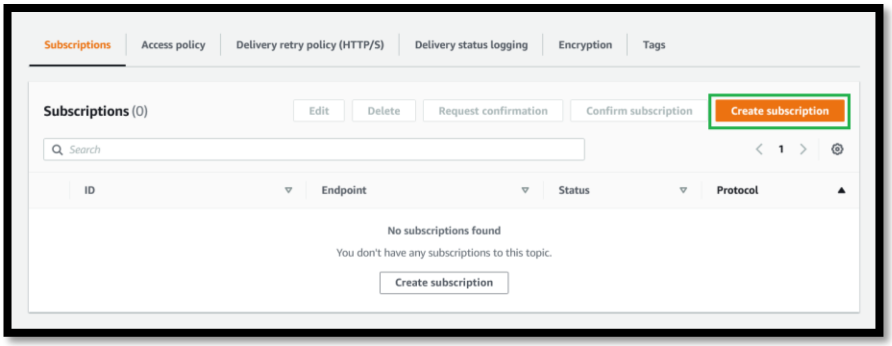
Warning

Details

Name Warning	Display name -
ARN arn:aws:sns:us-west-2:602914245248:Warning	Topic owner 602914245248
Type Standard	

Figure 74: Successful creation of the Warning topic

Scrolling down reveals the details of the Subscriptions to the topic, as shown in Figure 75. ‘Create subscription’ should be clicked to add details of the warning email recipient.

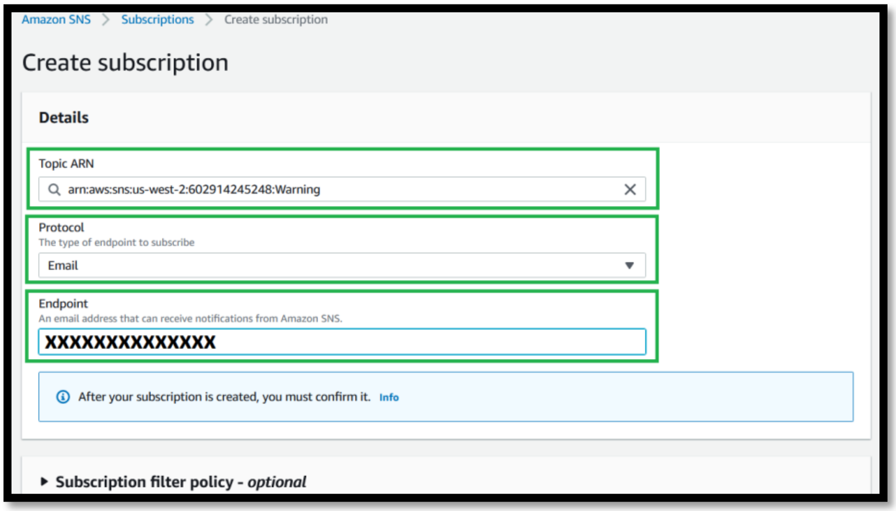


No subscriptions found

You don't have any subscriptions to this topic.

Figure 75: Warning topic subscriptions

Enter details as shown in Figure 76. 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.



Topic ARN: arn:aws:sns:us-west-2:602914245248:Warning

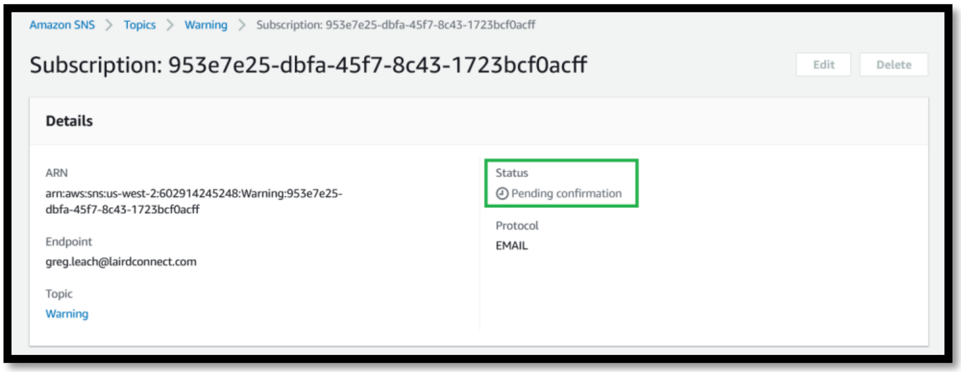
Protocol — The type of endpoint to subscribe: Email

Endpoint — An email address that can receive notifications from Amazon SNS.: XXXXXXXXXXXXXX

After your subscription is created, you must confirm it. Info

Figure 76: 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 77, until the subscription is verified.



Subscription: 953e7e25-dbfa-45f7-8c43-1723bcf0acff

ARN: arn:aws:sns:us-west-2:602914245248:Warning:953e7e25-dbfa-45f7-8c43-1723bcf0acff

Endpoint: greg.leach@lairdconnect.com

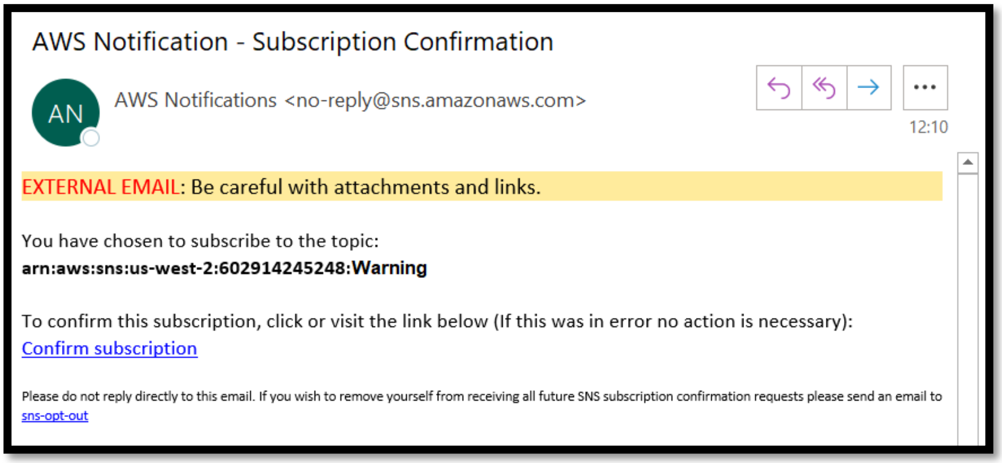
Topic: Warning

Status: Pending confirmation

Protocol: EMAIL

Figure 77: Pending confirmation of the Warning topic subscription

The verification email is as shown in Figure 78.


AWS Notification - Subscription Confirmation

AWS Notifications <no-reply@sns.amazonaws.com>

12:10

EXTERNAL EMAIL: Be careful with attachments and links.

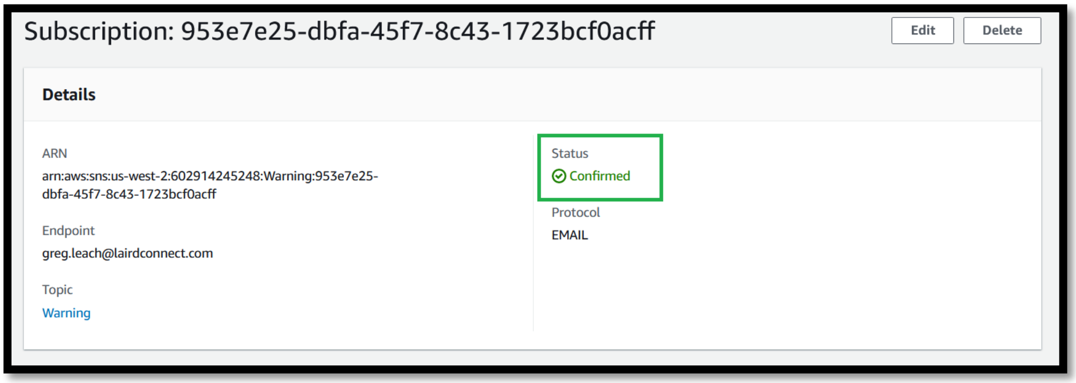
You have chosen to subscribe to the topic:
arn:aws:sns:us-west-2:602914245248:Warning

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
Confirm subscription

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to sns-opt-out

Figure 78: SNS subscription confirmation email

Having confirmed the subscription, the details appear as shown in Figure 79.


Subscription: 953e7e25-dbfa-45f7-8c43-1723bcf0acff

Details

ARN
arn:aws:sns:us-west-2:602914245248:Warning:953e7e25-dbfa-45f7-8c43-1723bcf0acff

Endpoint
greg.leach@lairdconnect.com

Topic
Warning

Status
Confirmed

Protocol
EMAIL

Figure 79: Confirmation of the Warning topic subscription

6.2.4 Creating the Warning Rule and topic and SNS Action

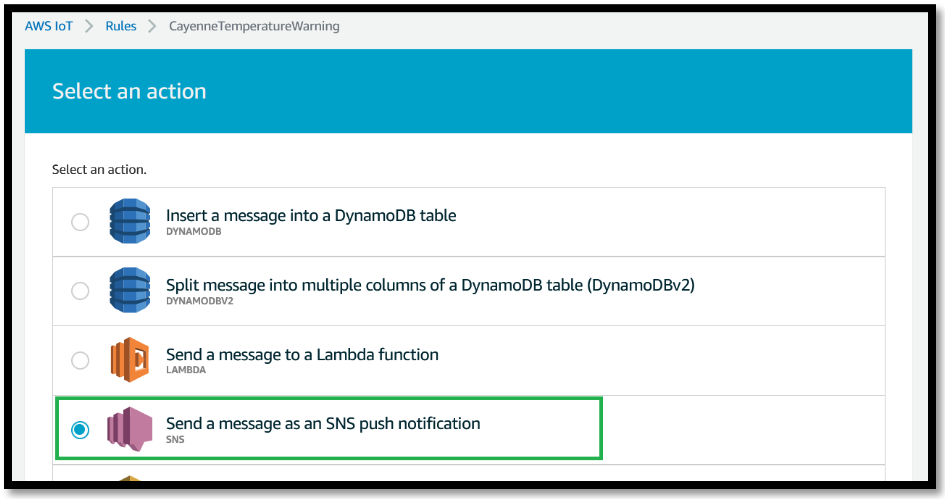
Create a second rule to only publish messages when the sensor temperature falls below 10˚C. Create the rule as described in section 6.1.4. The query is as follows:

```
SELECT * FROM 'Decoded' WHERE output.temperature < 10
```

This query ensures messages are only published when the temperature falls below 10˚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 an action to “Send a message as an SNS push notification” as well, as shown in Figure 80.



AWS IoT > Rules > CayenneTemperatureWarning

Select an action

Select an action.

○ Insert a message into a DynamoDB table
DYNAMODB

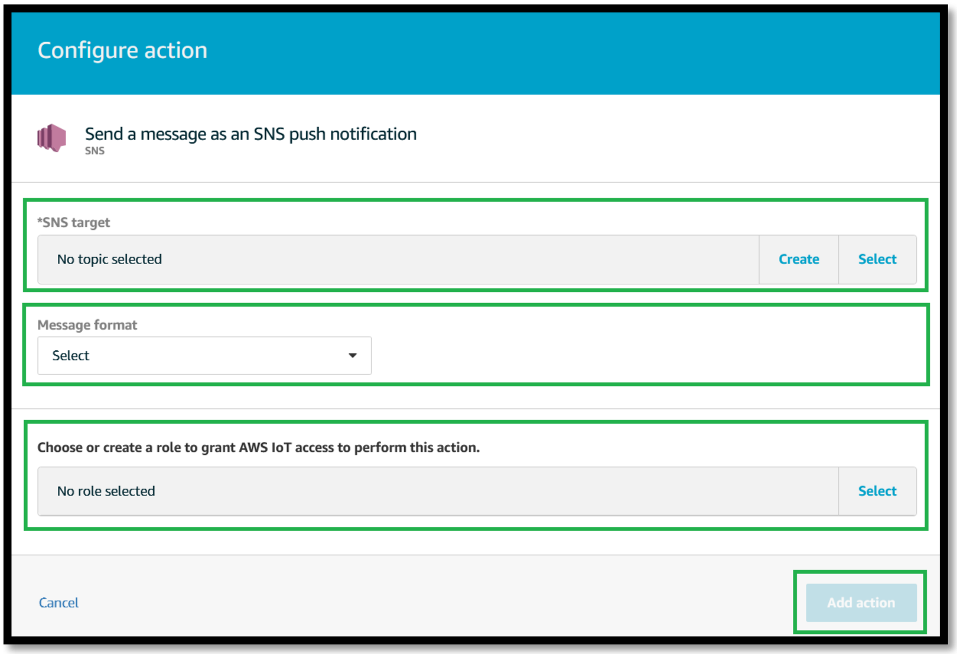
○ Split message into multiple columns of a DynamoDB table (DynamoDBv2)
DYNAMODBV2

○ Send a message to a Lambda function
LAMBDA

● Send a message as an SNS push notification
SNS

Figure 80: Adding the SNS push notification

When configuring the Action, set *SNS Target* to “Warning” and *Message format* to “Raw”. Create a Role for the Action and click **Add action** as shown in Figure 81.



Configure action

Send a message as an SNS push notification
SNS

*SNS target
No topic selected

Message format
Select

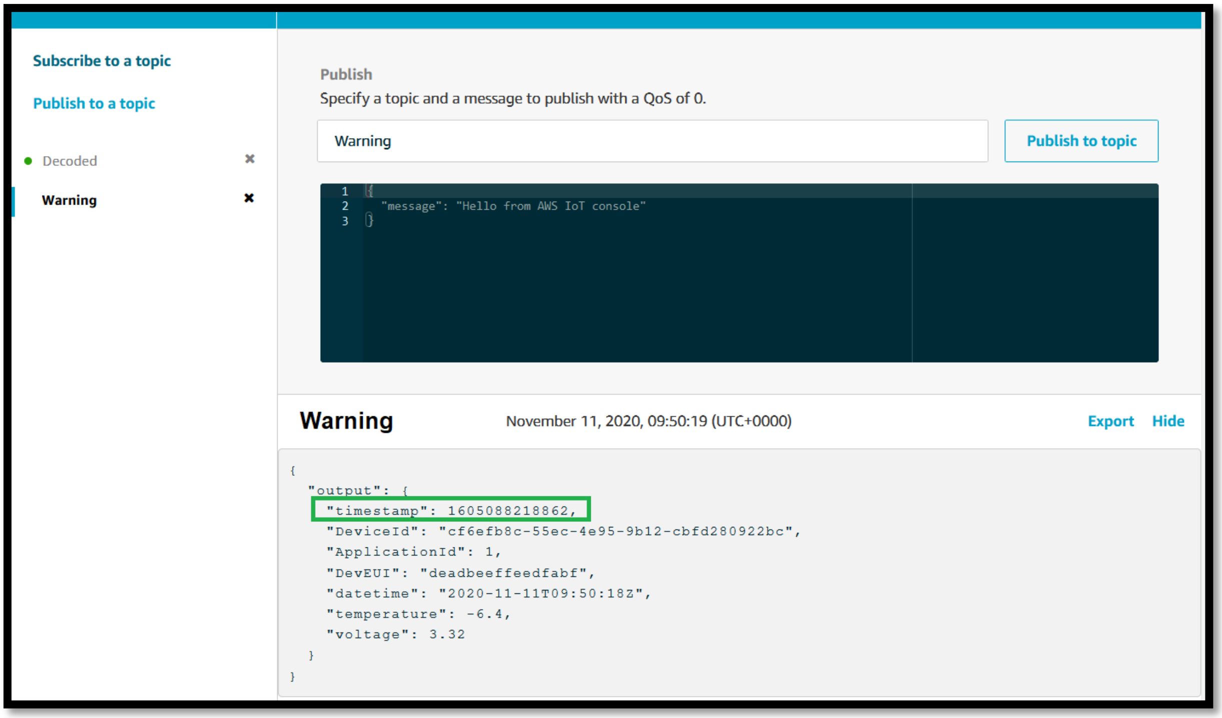
Choose or create a role to grant AWS IoT access to perform this action.
No role selected

Figure 81: Configuring the SNS push notification

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

6.2.5 Testing the application

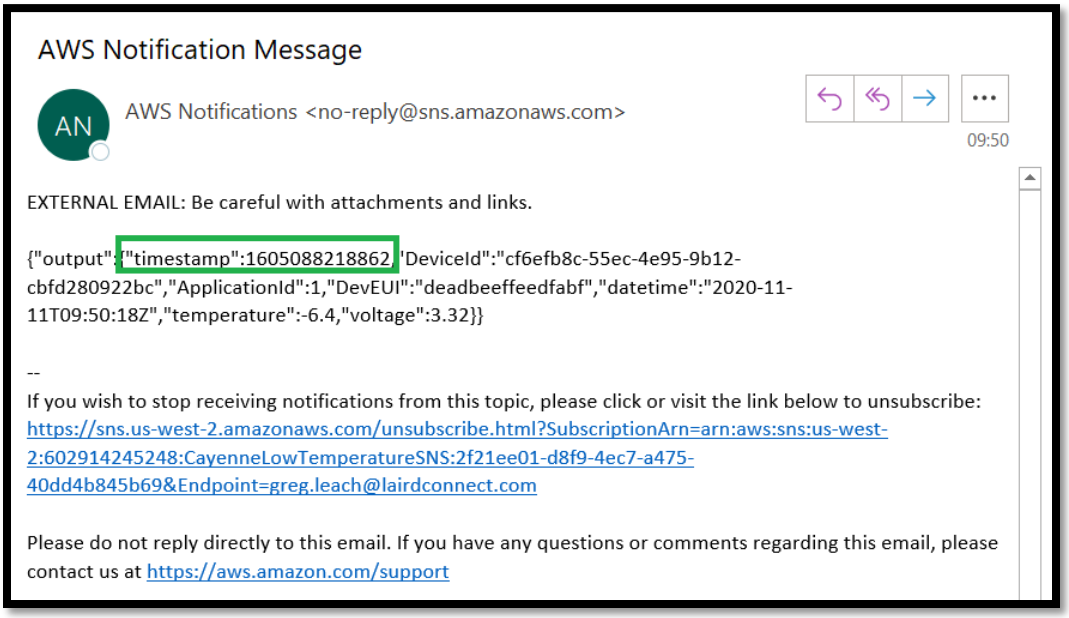
Use the AWS MQTT Client to subscribe to the “Decoded” and “Warning” topics. Messages are only published to the “Warning” topic in event of the temperature falling below 10˚C. For each message published to the ‘Warning’ topic as it’s shown in Figure 82, an email should be received as shown in Figure 83.



```
{
  "message": "Hello from AWS IoT console"
}
```

```
{
  "output": {
    "timestamp": 1605088218862,
    "DeviceId": "cf6efb8c-55ec-4e95-9b12-cbfd280922bc",
    "ApplicationId": 1,
    "DevEUI": "deadbeeffeedfabf",
    "datetime": "2020-11-11T09:50:18Z",
    "temperature": -6.4,
    "voltage": 3.32
  }
}
```

Figure 82: Message received by ‘Warning’ topic



AWS Notification Message

AWS Notifications <no-reply@sns.amazonaws.com>

EXTERNAL EMAIL: Be careful with attachments and links.

{"output":{"timestamp":1605088218862,"DeviceId":"cf6efb8c-55ec-4e95-9b12-cbfd280922bc","ApplicationId":1,"DevEUI":"deadbeeffeedfabf","datetime":"2020-11-11T09:50:18Z","temperature":-6.4,"voltage":3.32}}

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-west-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-west-2:602914245248:CayenneLowTemperatureSNS:2f21ee01-d8f9-4ec7-a475-40dd4b845b69&Endpoint=greg.leach@lairdconnect.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

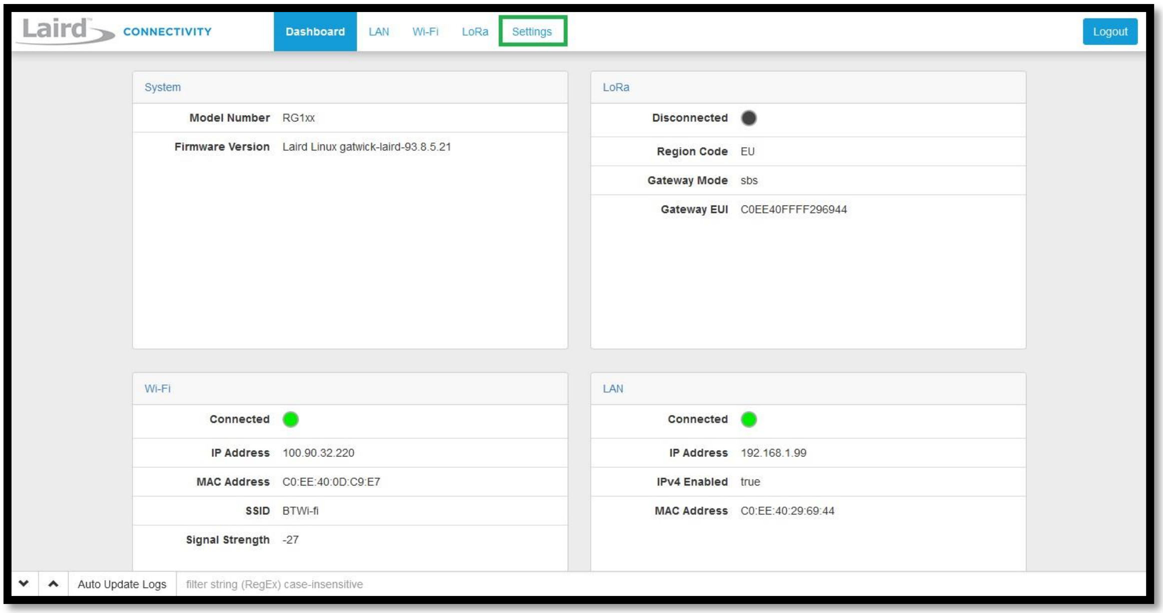
Figure 83: Email message received due to publish to ‘Warning’ topic

7 Gateway OTA Updates

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

7.1 Starting the firmware update

From the web interface Dashboard, click **Settings** as shown in Figure 84.



System

Model Number	RG1xx
Firmware Version	Laird Linux gatwick-laird-93.8.5.21

LoRa

Disconnected	●
Region Code	EU
Gateway Mode	sbs
Gateway EUI	C0EE40FFFF296944

Wi-Fi

Connected	●
IP Address	100.90.32.220
MAC Address	C0:EE:40:0D:C9:E7
SSID	BTWi-fi
Signal Strength	-27

LAN

Connected	●
IP Address	192.168.1.99
IPv4 Enabled	true
MAC Address	C0:EE:40:29:69:44

Figure 84: Opening the gateway Settings page

From the Settings page, click **Update Firmware** as shown in Figure 85.

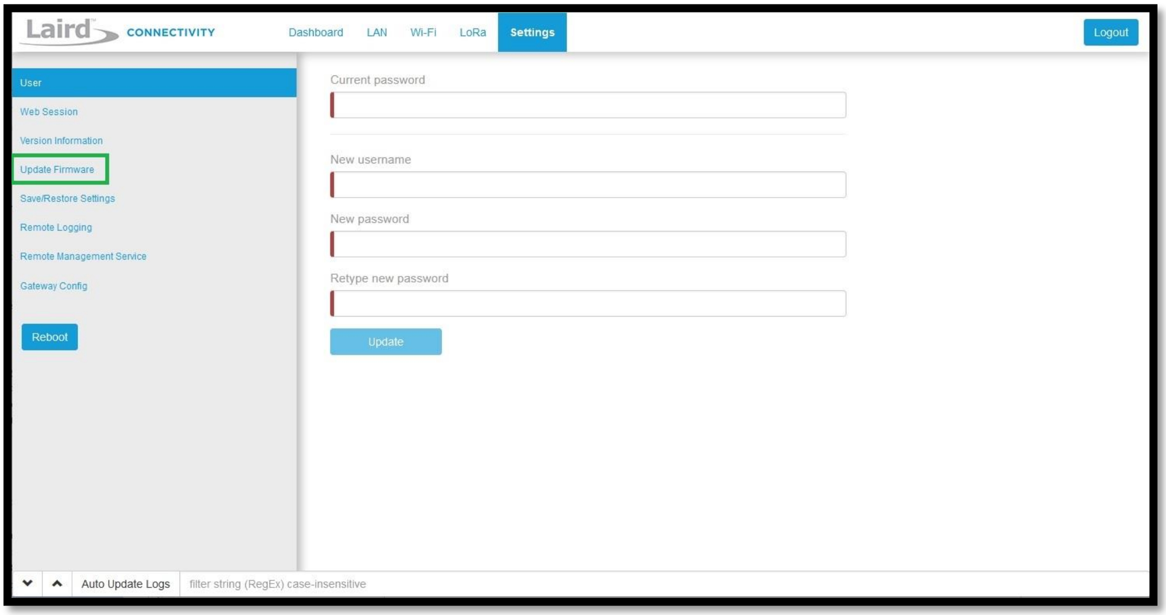


Figure 85: Opening the Update Firmware page

The Update Firmware page is shown in Figure 86.

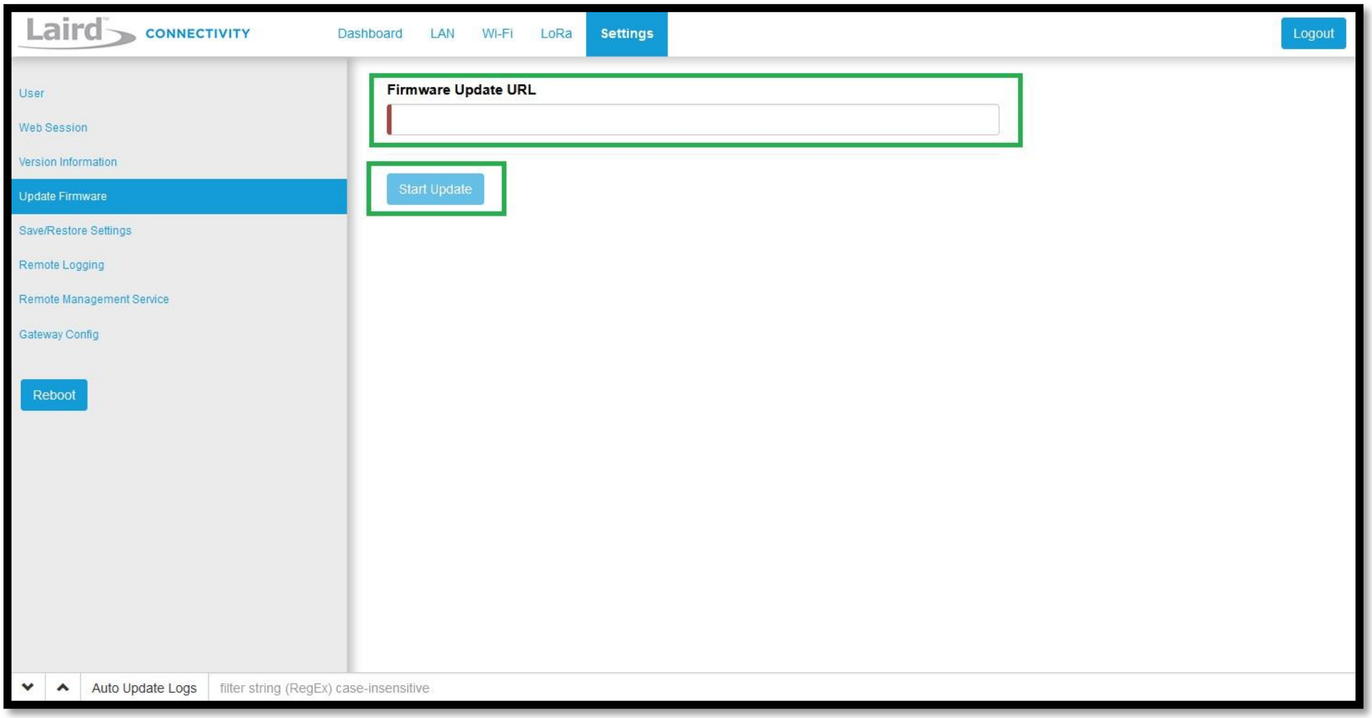


Figure 86: Update Firmware page

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.

7.2 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**.

7.2.1 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.

7.2.2 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.

7.2.3 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.

7.2.4 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.

7.2.5 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.

7.2.6 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

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

This is the latest production release.

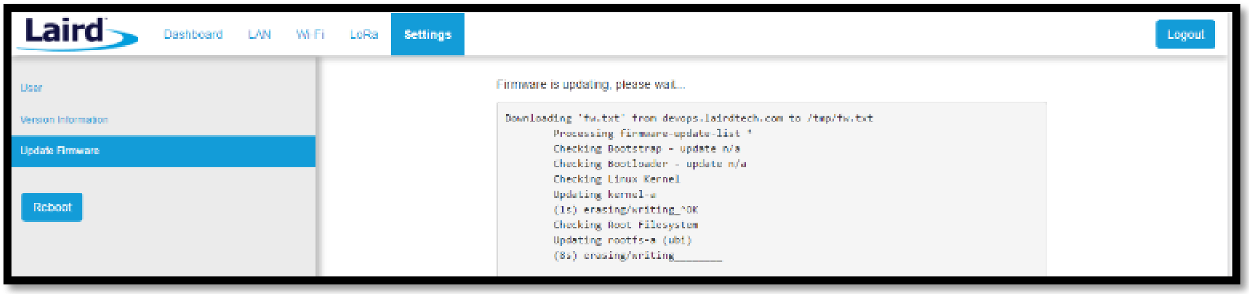
7.2.8 Firmware Version 93.8.5.25 (GA5.2)

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

7.3 Firmware update process

Click **Start** Update. Details of the update process appear on the Update Firmware page as shown in Figure 87.

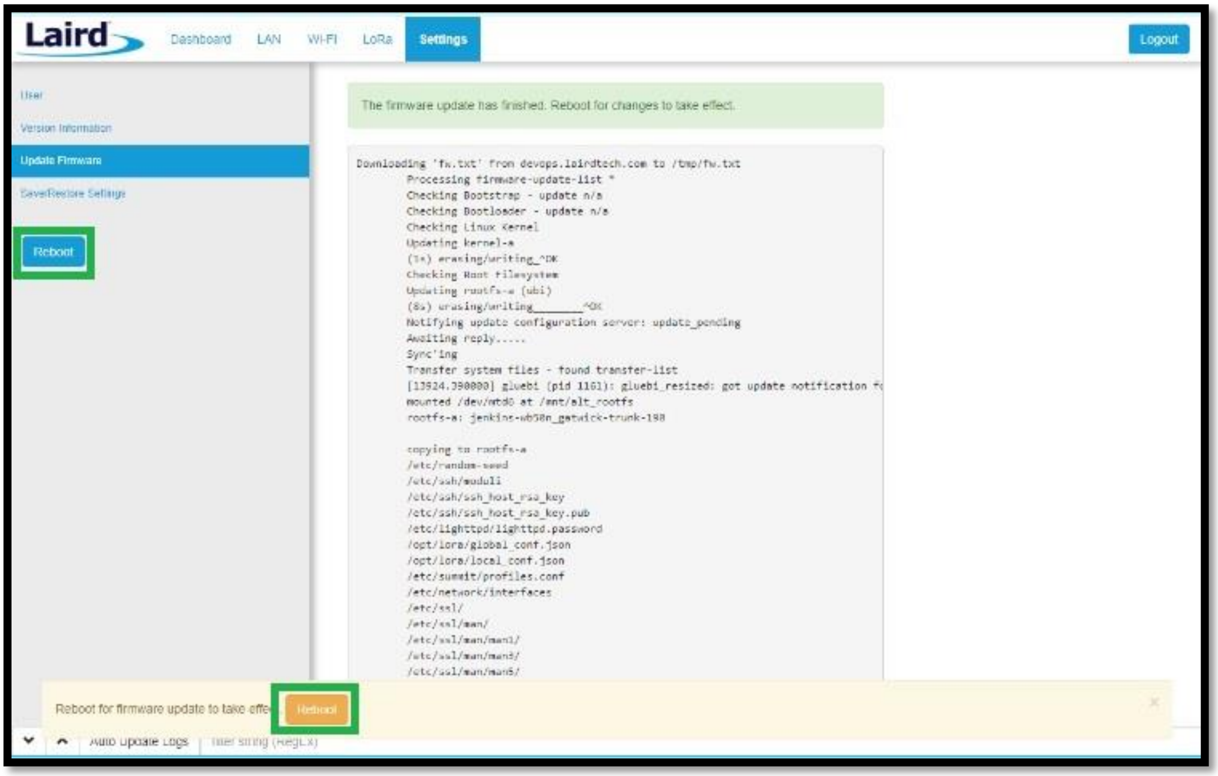


Firmware is updating, please wait...

```
Downloading `fw.txt` from devops.lairdtech.com to /tmp/fw.txt
        Processing firmware-update-list *
        Checking Bootstrap - update n/a
        Checking Bootloader - update n/a
        Checking Linux Kernel
        Updating kernel-a
        (1s) erasing/writing_^OK
        Checking Root Filesystem
        Updating rootfs-a (ubi)
        (8s) erasing/writing________
```

Figure 87: Firmware update progress display

Upon completion of the update, the page prompts you to reboot the gateway as shown in Figure 88.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.



The firmware update has finished. Reboot for changes to take effect.

```
Downloading 'fw.txt' from devops.lairdtech.com to /tmp/fw.txt
        Processing firmware-update-list *
        Checking Bootstrap - update n/a
        Checking Bootloader - update n/a
        Checking Linux Kernel
        Updating kernel-a
        (1x) erasing/writing_^OK
        Checking Root filesystem
        Updating rootfs-a (ubi)
        (8x) erasing/writing________^OK
        Notifying update configuration server: update_pending
        Awaiting reply.....
        Sync'ing
        Transfer system files - found transfer-list
        [13924.390000] gluebi (pid 1161): gluebi_resized: got update notification fo
        mounted /dev/mtd0 at /mnt/alt_rootfs
        rootfs-a: jenkins-wb50n_gatwick-trunk-198

        copying to rootfs-a
        /etc/random-seed
        /etc/ssh/moduli
        /etc/ssh/ssh_host_rsa_key
        /etc/ssh/ssh_host_rsa_key.pub
        /etc/lighttpd/lighttpd.password
        /opt/lora/global_conf.json
        /opt/lora/local_conf.json
        /etc/summit/profiles.conf
        /etc/network/interfaces
        /etc/ssl/
        /etc/ssl/man/
        /etc/ssl/man/man1/
        /etc/ssl/man/man3/
        /etc/ssl/man/man5/
```

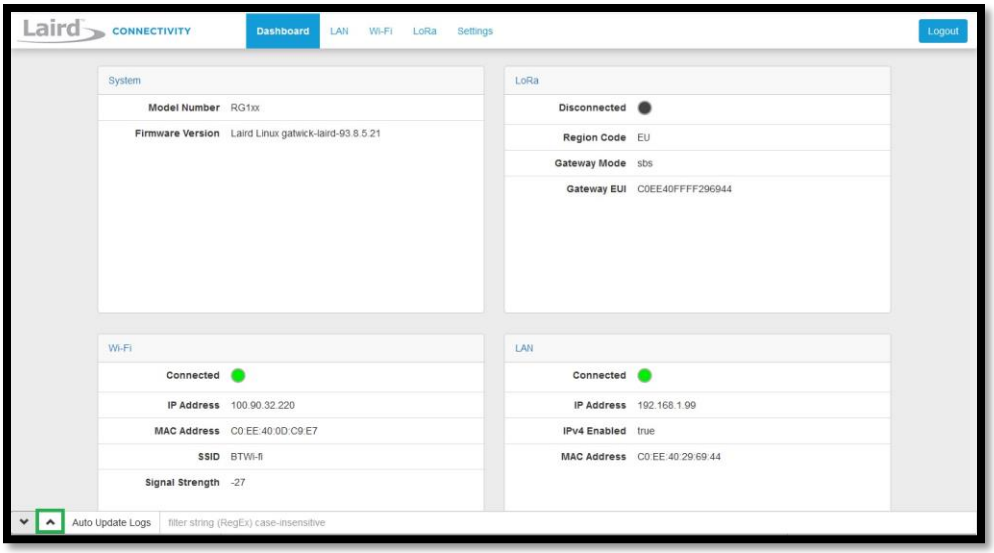
Reboot for firmware update to take effe

Figure 88: Reboot prompt following firmware update

8 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 89 to partially reveal the log window. Clicking again will further reveal the window.



System
Model Number RG1xx
Firmware Version Laird Linux gatwick-laird-93.8.5.21

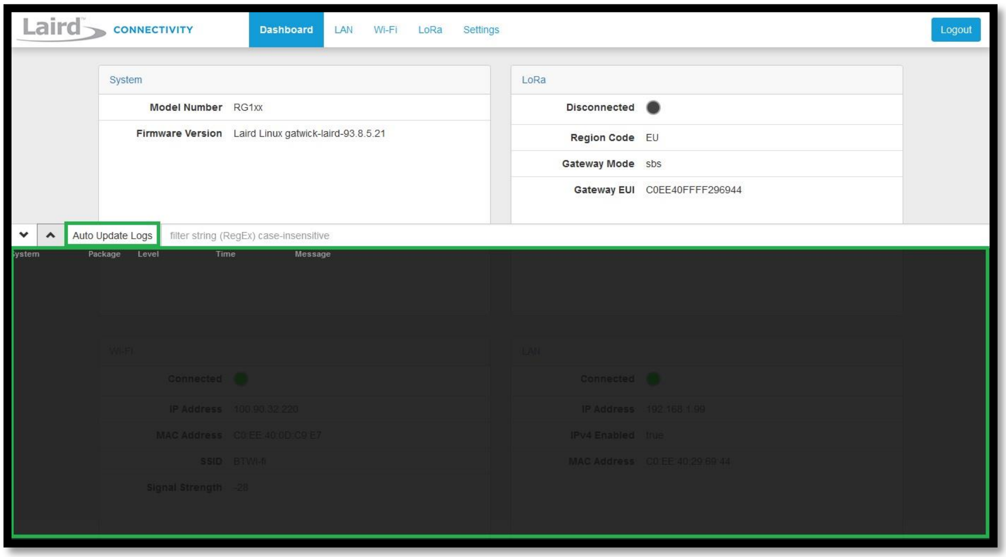
LoRa
Disconnected ●
Region Code EU
Gateway Mode sbs
Gateway EUI C0EE40FFFF296944

Wi-Fi
Connected ●
IP Address 100.90.32.220
MAC Address C0:EE:40:0D:C9:E7
SSID BTWI-fi
Signal Strength -27

LAN
Connected ●
IP Address 192.168.1.99
IPv4 Enabled true
MAC Address C0:EE:40:29:69:44

Figure 89: Enabling the gateway log

Click **Auto Update Logs** to auto-refresh the log as shown in Figure 90.



System
Model Number RG1xx
Firmware Version Laird Linux gatwick-laird-93.8.5.21

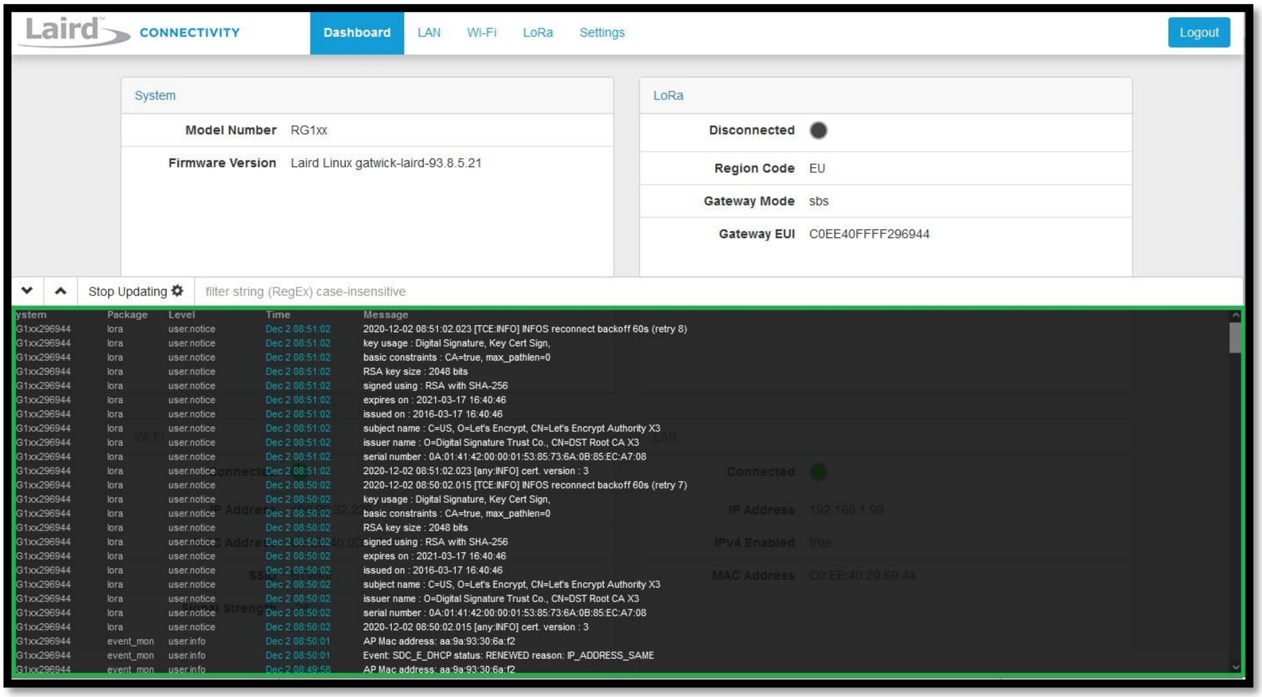
LoRa
Disconnected ●
Region Code EU
Gateway Mode sbs
Gateway EUI C0EE40FFFF296944

Wi-Fi
Connected ●
IP Address 100.90.32.220
MAC Address C0:EE:40:0D:C9:E7
SSID BTWI-fi
Signal Strength -28

LAN
Connected ●
IP Address 192.168.1.99
IPv4 Enabled true
MAC Address C0:EE:40:29:69:44

Figure 90: Gateway log window

The log window is continuously updated with details of activities being performed by the gateway as shown in Figure 91.



System

Model Number RG1xx

Firmware Version Laird Linux gatwick-laird-93.8.5.21

LoRa

Disconnected ●

Region Code EU

Gateway Mode sbs

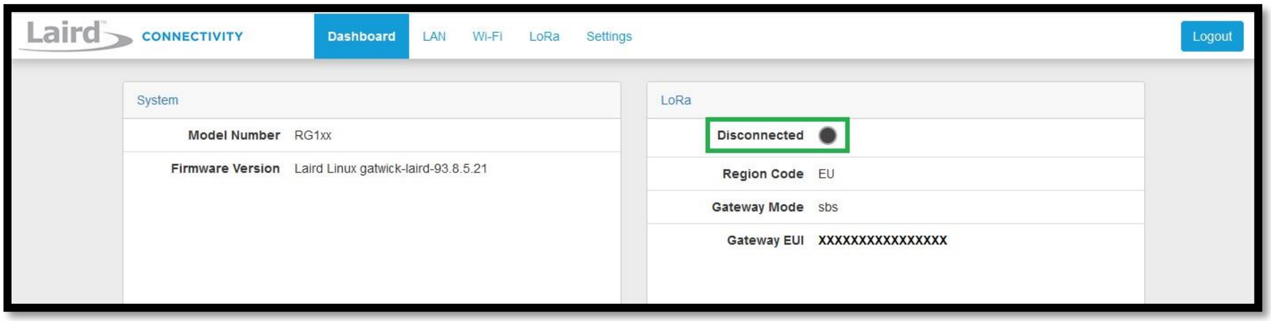
Gateway EUI C0EE40FFFF296944

```
ystem        Package    Level        Time             Message
G1xx296944   lora       user.notice  Dec 2 08:51:02   2020-12-02 08:51:02.023 [TCE:INFO] INFOS reconnect backoff 60s (retry 8)
G1xx296944   lora       user.notice  Dec 2 08:51:02   key usage : Digital Signature, Key Cert Sign,
G1xx296944   lora       user.notice  Dec 2 08:51:02   basic constraints : CA=true, max_pathlen=0
G1xx296944   lora       user.notice  Dec 2 08:51:02   RSA key size : 2048 bits
G1xx296944   lora       user.notice  Dec 2 08:51:02   signed using : RSA with SHA-256
G1xx296944   lora       user.notice  Dec 2 08:51:02   expires on : 2021-03-17 16:40:46
G1xx296944   lora       user.notice  Dec 2 08:51:02   issued on : 2016-03-17 16:40:46
G1xx296944   lora       user.notice  Dec 2 08:51:02   subject name : C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
G1xx296944   lora       user.notice  Dec 2 08:51:02   issuer name : O=Digital Signature Trust Co., CN=DST Root CA X3
G1xx296944   lora       user.notice  Dec 2 08:51:02   serial number : 0A:01:41:42:00:00:01:53:85:73:6A:0B:85:EC:A7:08
G1xx296944   lora       user.notice  Dec 2 08:51:02   2020-12-02 08:51:02.023 [any:INFO] cert. version : 3
G1xx296944   lora       user.notice  Dec 2 08:50:02   2020-12-02 08:50:02.015 [TCE:INFO] INFOS reconnect backoff 60s (retry 7)
G1xx296944   lora       user.notice  Dec 2 08:50:02   key usage : Digital Signature, Key Cert Sign,
G1xx296944   lora       user.notice  Dec 2 08:50:02   basic constraints : CA=true, max_pathlen=0
G1xx296944   lora       user.notice  Dec 2 08:50:02   RSA key size : 2048 bits
G1xx296944   lora       user.notice  Dec 2 08:50:02   signed using : RSA with SHA-256
G1xx296944   lora       user.notice  Dec 2 08:50:02   expires on : 2021-03-17 16:40:46
G1xx296944   lora       user.notice  Dec 2 08:50:02   issued on : 2016-03-17 16:40:46
G1xx296944   lora       user.notice  Dec 2 08:50:02   subject name : C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
G1xx296944   lora       user.notice  Dec 2 08:50:02   issuer name : O=Digital Signature Trust Co., CN=DST Root CA X3
G1xx296944   lora       user.notice  Dec 2 08:50:02   serial number : 0A:01:41:42:00:00:01:53:85:73:6A:0B:85:EC:A7:08
G1xx296944   lora       user.notice  Dec 2 08:50:02   2020-12-02 08:50:02.015 [any:INFO] cert. version : 3
G1xx296944   event_mon  user.info    Dec 2 08:50:01   AP Mac address: aa:9a:93:30:6a:f2
G1xx296944   event_mon  user.info    Dec 2 08:50:01   Event: SDC_E_DHCP status: RENEWED reason: IP_ADDRESS_SAME
G1xx296944   event_mon  user.info    Dec 2 08:49:58   AP Mac address: aa:9a:93:30:6a:f2
```

Figure 91: Update of the gateway log window

9 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 92.



System

Model Number RG1xx

Firmware Version Laird Linux gatwick-laird-93.8.5.21

LoRa

Disconnected ●

Region Code EU

Gateway Mode sbs

Gateway EUI XXXXXXXXXXXXXXXX

Figure 92: 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

10 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
[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