TEKTELIC COMMUNICATIONS INC.

KONA GW AND SENSOR WITH AWS IOT CORE FOR LORAWAN GETTING STARTED GUIDE

Document type: Getting Started Guide
Document status: Final
Last update: 2020-12-18

Table of Contents

1 Document Information

1.1 Naming Conventions

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

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

Revision: v0.3
Date: Dec. 18, 2020
Description of the change: Updated Section 4.4, 6.4.3 and Section 5.

2 Overview

Introducing the Developer Starter Kit containing the Versatile LoRaWAN® Smart Room Sensor and the Highly Scalable KONA Micro Gateway. The KONA Smart Room Sensor integrates practical functionality into a very small form factor. The Smart Room Sensor is an ideal solution for holistically monitoring the home and office environment. The device is capable of measuring and reporting temperature, humidity, light, movement, motion, shock, detecting leaks, open / closed doors and windows. It also supports battery status updates for easy maintenance. Paired with the KONA Micro IoT Gateway, which is designed for enterprise and lightweight industrial applications that require “Always On” connectivity. Configured with an internal 3G/4G modem and a built-in battery backup, the KONA Micro IoT gateway continues to operate and transmits sensor data to the network even when the main site has lost power.

3 Hardware Description

3.1 DataSheet

KONA Micro Gateway: https://tektelic.com/uploads/Brochures/Kona%20Micro.pdf
Smart Room Sensor Base & PiR : https://tektelic.com/uploads/Brochures/Smart%20Room%20Sensor.pdf

3.2 Standard Kit Contents

Contents of standard shipping hardware package contain:

1. KONA Micro Gateway
2. Power Adapter
3. Ethernet Cable
4. LoRa Antenna
5. Smart Room Sensor Base

3.3 Smart Room Sensor PiRUser Provided items

Power Source for Gateway

3.4 3rd Party purchasable items

None

3.5 Additional Hardware References

None

4 Setup your AWS account and Permissions

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

4.1 Overview

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

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

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

4.2 Set up Roles and Policies in IAM

4.2.1 Add an IAM Role for CUPS server

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

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

- Go to the IAM Roles page on the IAM console
- Choose **Create role**.
- On the **Create Role** page, choose **Another AWS account**.
- For **Account ID**, enter your account id.
- Choose **Next: Permissions**
- In the search box next to **Filter policies**, enter *AWSIoTWirelessGatewayCertManager*.
    - If the search results show the policy named *AWSIoTWirelessGatewayCertManager*, select it by clicking on the checkbox.
    - If the policy does not exist, please create it as follows:
        - Go to the IAM console
        - Choose **Policies** from the navigation pane.
        - Choose **Create Policy**. Then choose the **JSON** tab to open the policy editor. Replace the existing template with this trust policy document:

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

        - Choose **Review Policy** to open the *Review* page.
        - For **Name**, enter *AWSIoTWirelessGatewayCertManager*. **Note** that you must enter the name as AWSIoTWirelessGatewayCertManager and must not use a different name. This is for consistency with future releases.
        - For **Description**, enter a description of your choice.

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

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

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

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

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

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

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. First, create an IAM role with permissions to describe the IoT end point and to deliver messages to IoT cloud. Then, update the trust policy to grant AWS IoT Core for LoRaWAN permission to assume this IAM role when delivering messages from devices to your account.

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

First, create a policy with the permissions described above.

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

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action":
          [
```

```
                "iot:DescribeEndpoint",
                "iot:Publish"
            ],
            "Resource": "*"
        }
    ]
}
```

- Choose **Review Policy** to open the Review page. For Name, enter a name of your choice. For **Description**, enter a description of your choice.
- Choose **Create policy**.

Now, create a role that will use the above policy.

- In the IAM console, choose **Roles** from the navigation pane to open the **Roles** page.
- Choose **Create Role**.
- In **Select type of trusted entity**, choose **Another AWS account**.
- In **Account ID**, enter your AWS account ID, and then choose **Next: Permissions**.
- Choose **Next: Permissions**
- Search for your IAM policy created in the step above. Type in the policy name to find your policy. Select it.
- Choose **Next: Tags**.
- Choose **Next: Review** to open the Review page. For **Role name**, enter an appropriate name of your choice. For **Description**, enter a description of your choice.
- Choose **Create role**.

Update your policy's trust relationship.

- 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
- Choose the **Trust relationships** tab to navigate to the Trust relationships page.
- Choose **Edit trust relationship**. The principal AWS role in your trust policy document defaults to root. 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**

4.3 Add the Gateway to AWS IoT

4.3.1 Preparation

To complete setting up your gateway, you need:

- LoRaWAN region. For example, if the gateway is deployed in a US region, the gateway must support LoRaWAN region US915.

- Gateway LNS-protocols. Currently, the LoRa Basics Station protocol is supported.
- Gateway ID (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.
- Your gateway's Basics Station version must be 2.0.5 or higher.

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 (**console.aws.amazon.com/iot)** and login.
- Select **Wireless connectivity** in the navigation panel on the left.
- Choose **Intro**, and then choose **Get started**. This step is needed to pre-populate the default profiles.
- Under **Add LoRaWAN gateways and wireless devices**, choose **Add gateway**.
- In the **Add gateway** section, fill in the **GatewayEUI** (found on the bottom of your gateway as GW ID) and **Frequency band (RF Region)** fields.
- Enter a descriptive name in the **Name – optional** field. We recommend that you use the Gateway EUI as the name.
- Choose **Add gateway**
- On the **Configure your Gateway** page, find the section titled **Gateway certificate**.
- Select **Create certificate**.
- Once the **Certificate created and associated with your gateway** message is shown, select **Download certificates** to download the certificate (xxxxx.cert.pem) and private key (xxxxxx.private.key). We recommend that you store all the downloaded files in the same folder.
    - Then rename xxxx.cert.pem file to cups.crt and xxxx.private.key to cups.key.
    - Create a copy of *cups.key* and name it *tc.key*.
    - Create a copy of *cups.crt* and name it *tc.crt*.
- In the section **Provisioning credentials**, choose **Download server trust certificates** to download the CUPS (cups.trust) and LNS (lns.trust) server trust certificates.
    - Keep the cups.trust file as it is.
    - Rename the lns.trust file to tc.trust.
- Copy the CUPS and LNS endpoints and save them for use while configuring the gateway.
    - Create cups.uri file with CUPS Endpoint URL:
      e.g: https://EXAMPLE.cups.lorawan.REGION.amazonaws.com:443
    - Create tc.uri file with LNS Endpoint URL:
      e.g: wss://EXAMPLE.gateway.lorawan.REGION.amazonaws.com:443

Make sure that you have the following 8 files from the steps above as you'll need them to configure your gateway:

- tc.uri
- tc.trust
- tc.key
- tc.crt
- cups.uri
- cups.trust
- cups.key
- cups.crt

- Choose **Submit** to add the gateway.

4.4 Add a LoRaWAN Device to AWS IoT

4.4.1 Preparation

Locate and note the following specifications about your endpoint device.

- LoRaWAN region. This must match the gateway LoRaWAN region. The following Frequency bands (RF regions) are supported:
    - EU868

    - US915
    - EU433
- MAC Version. This must be one of the following:
    - V1.0.2
    - v1.0.3
    - v1.1
- OTAA v1.0x and OTAA v1.1 are supported.
- ABP v1.0x and ABP v1.1 are supported.

Locate and note the following information from your device manufacturer:

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

Note: Tektelic Devices support v1.0.2. So, If you are using Tektelic Device choose “OTAA v1.0.x” from the list.

4.4.2 Verify Profiles

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

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

- Navigate to the AWS IoT Core console. In the navigation pane, choose **Wireless connectivity.**
- In the navigation pane, choose **Profiles**
- In the **Device Profiles** section, there are some pre-defined profiles listed.
- Check each of the profiles to determine if one of them will work for you.
- If not, select **Add device profile** and set up the parameters as needed. For US 915 as an example, the values are:
    - MacVersion 1.0.3
    - RegParamsRevision RP002-1.0.1
    - MaxEirp 10
    - MaxDutyCycle 10
    - RfRegion US915
    - SupportsJoin true
- Continue once you have a device profile that will work for you.
- In the **Service Profiles** section, there are some pre-defined profiles listed. Check each of the profiles to determine if one of them will work for you.
- If not, select **Add service profile** and set up the parameters as needed. As an example, the default service profile parameters are shown below. However, only the AddGwMetadata setting can be changed at this time.
    - UlRate 60
    - UlBucketSize 4096
    - DlRate 60
    - DlBucketSize 4096
    - AddGwMetadata true
    - DevStatusReqFreq 24
    - DrMax 15
    - TargetPer 5
    - MinGwDiversity 1

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

4.4.3 Set up a Destination for device traffic

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

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

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

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

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

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

4.4.4 Register the Device

All the Tektelic Devices support v1.0.2.
Now register an endpoint device with AWS IoT Core for LoRaWAN as follows:

- Go to the AWS IoT Core console.
- Select **Wireless connectivity** in the navigation panel on the left.
- Select **Devices**
- Choose **Add wireless device**
- On the **Add device** page, select the LoRaWAN specification version in the drop-down under **Wireless device specification**.
  - Tektelic Devices support v1.0.2. So, If you are using Tektelic Device choose "OTAA v1.0.x" from the list.
- 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.
  - v1.0.2 supports only Appkey.
- Enter a name for your device in the **Wireless device name – optional** field.
- In the **Profiles** section, under **Wireless device profile**, find a drop-down option that corresponds to your device and region.
  - NOTE: Compare your device details to ensure the device profile is correct. If there are no valid default options, you will have to create a new profile (see the section Verify Profiles).
- Choose **Next**
- Choose the destination you created earlier (*ProcessLoRa*) from the drop-down under **Choose destination.**
- Choose **Add device**
  You will see a message saying "*Wireless device added*", indicating that your device has been set up successfully.

5 Set up and Configure the Gateway

5.1 Set up Gateway hardware

KONA Micro Gateway Unboxing

KONA Micro Gateway Setup steps.
Box contains:

- KONA Micro Gateway
- Power Adapter
- Ethernet Cable
- LoRa Antenna
- Smart Room Sensor Base
- Smart Room Sensor PiR

Setup:

- Remove items from box
- Connect LoRa Antenna to Micro Gateway
- Plug into power source
- Plug into ethernet source

Your KONA Micro is now live and ready to connect!

Detailed Quick start guides are available for KONA Micro Gateway and Smart Room Sensors at
support@tektelic.com
https://support.tektelic.com/portal/en/kb/support

5.2 Set up Gateway Software

The minimum BSP version is required for this is, For Mega and Macro, BSP Version should be 4.x.x and for Micro, BSP version should be 3.x.x.

- Login to your Gateway using SSH. By default, user name is “root” and the password is “Gateway’s 9-digit serial number” (You can find this information on the label on your Gateway)

You can check the BSP version on your gateway by issuing “system_version” command on the Gateway’s console (using SSH).

5.2.1 Preparing Basic Station for the installation on BSP 3.0.x and 3.1.x (Micro) and 4.0.x and 4.1.x (Mega and Macro):

- Please create an account on our support portal (https://support.tektelic.com/portal/en/signin) and go to knowledge base -> Basic Station
- Download the Basic Station Package. (Basic-Station-packages-vx.x.x-for-Tektelic-gateways.tar.gz)
- Then, upload the Basic-Station-packages-vx.x.x-for-Tektelic-gateways.tar.gz to the directory /lib/firmware on the target gateway and extract it using following command.

```
tar -C /lib/firmware \
    -zxvf /lib/firmware/Basic-Station-packages-vx.x.x-for-Tektelic-gateways.tar.gz
```

- Add the feed location to the package manager configuration file by using following command:

```
echo "src/gz bstn file:///lib/firmware/Basic-Station-packages-vx.x.x-for-Tektelic-gateways" \
    > /etc/opkg/bstn-feed.conf
```

- Then enter the following command.

```
opkg update
```

5.2.1.1 Installing Basic Station packages using command line:

To install the basic Station packages, run the following command:

- opkg install tektelic-bstn curl libcurl4

5.2.2 Preparing Basic Station for the installation on Micro BSP 3.2.x, Mega/Macro 4.2.x or later:

- **Obtain the ipk/bsp package by contacting our support team.** Please create an account on our support portal (https://support.tektelic.com/portal/en/signin) or send an email to support@tektelic.com to contact us.
- Upload the ipk/bsp folder to the gateway and extract it into /lib/firmware.
- Add the feed location to the package manager configuration file by running following command.
  echo "src/gz bstn file:///lib/firmware/bsp" > /etc/opkg/bstn-feed.conf
- Then enter the following command.
  opkg update

5.2.2.1 Installing Basic Station packages using command line:

To install the basic Station packages, run the following command:

- opkg install tektelic-bstn curl libcurl4

Note:
Please create an account on our support portal (https://support.tektelic.com/portal/en/signin) and go to knowledge base for Gateway and Device Guides and Documentation.
To get up to date information about our new BSP Releases for Gateways and Firmware releases for Devices, please go to community in our support portal and select "FOLLOW" button under TEKTELIC announcements. Then you will receive email notifications whenever we release new software.

If you have any questions or issues reach out to support@tektelic.com, one of our Customer Support Specialists will assist you.

5.3 Additional Software References

None

5.4 Configure the Gateway

This configuration is applicable for Kona Mega, Kona Macro and Kona Micro gateways.

a. Login to your Gateway using SSH. By default, user name is “root” and the password is “Gateway’s 9-digit serial number” (You can find this information on the label on your Gateway)

b. Make sure Basic Station and Packet Forwarder are installed.

- To check whether the packet forwarder is installed, enter “system_version” command on the console and look for Packet Forwarder, if it is listed then which means packet forwarder is installed.

```
root@kona-micro:~# system_version

Distributor ID:       Tektelic
Description:          Tektelic Kona Micro GNU/Linux 3.1.3
Release:              3.1.3

Product:              Kona Micro
u-boot:               2013.07-rc2-kona-micro-indoor-v0.7-gd941e52ab1
Linux kernel:         3.12.17-tektelic-2.3.0-kona-micro-indoor-g56fb3ac90d

System monitor:       tektelic-system-monitor-0.15-r6
SNMP agent:           tektelic-snmp-agents-1.1.0-r12
LTE connection mgr:   modem-connection-manager-0.42-r8
Network monitor:      kona-network-monitor-0.19-r7
NS switcher:          kona-ns-switcher-0.36-r13
Packet forwarder:     kona-pkt-forwarder-4.0.22-r121
LoRa HAL:             tektelic-lora-hal-3.6.1-r2
BIST manager:         tektelic-bist-manager-0.7-r4
BSP upgrade tool:     tektelic-upgrade-1.4.2-r30.p17
Backup tool:          tektelic-backup-1.5.1-r16
FPGA access tool:     tektelic-fpga-access-1.0.0-r7
TCS agent:            tektelic-tcs-1.3.2-r47

GPIO FPGA:            5007 build 0027
root@kona-micro:~#
```

- To check whether the Basic Station is installed, enter “opkg list-installed | grep bstn” command on the console.

```
root@kona-micro:~#
root@kona-micro:~# opkg list-installed | grep bstn
tektelic-bstn - 1.4.1-r58
root@kona-micro:~#
```

- If they are not installed, please reach out to us on our support portal (sign up required - https://support.tektelic.com/portal/en/signin), or support email – support@tektelic.com

c. Then make sure Basic Station and Packet Forwarder are running.

- To check whether they are running, enter “ps aux | grep pkt” and “ps aux | grep bstn” command on the console, if they both show up with process id which means they both are running.
  For Packet Forwarder:

```
root@kona-mega:~# ps aux | grep pkt
pktfwd    1981  5.3  0.4  63276  2428 ?        Sl   01:21  57:18 /usr/bin/pkt_forwarder -c /etc/default/config.json -s
root     10366  0.0  0.0   1804   460 ttyO0    S+   19:10   0:00 grep pkt
```

  For Basic Station:

```
root@kona-mega:~# ps aux | grep bstn
root      2010  0.1  0.7  24388  3684 ?        SLl  01:22   1:40 /usr/sbin/tek_bstn -c /etc/default/bstn.toml -v 3
root     11161  0.0  0.0   1804   464 ttyO0    S+   19:10   0:00 grep bstn
```

- If they are not running, please reach out to us on our support portal (sign up required - https://support.tektelic.com/portal/en/signin), or support email – support@tektelic.com

d. Configure Packet Forwarder:

- Update the “server address” in /etc/default/config.json file to 127.0.0.1. Then restart the packet forwarder. (/etc/init.d/pkt_fwd restart)

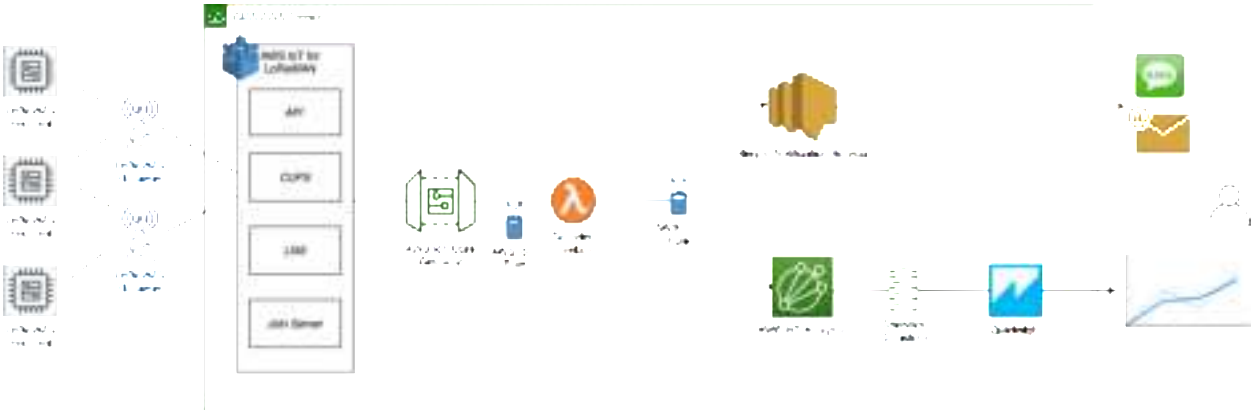
e. Configure Basic Station:
   - Keys and certificates required:
     - tc.uri
     - tc.trust
     - tc.key
     - tc.crt
     - cups.uri
     - cups.trust
     - cups.key
     - cups.crt
   - Copy the previously downloaded and created keys and certificates (see section 4.3.2). and put them into /etc/bstn directory on your GW. (You can use winscp to transfer files from windows PC)
   - By default, CUPS is enabled in the Basic Station to connect with Server. If you don't want to use CUPS then you can disable that by set "skip_cups=true" in /etc/default/bstn.toml file.
   - Then enter the following command to restart the Basic Station.
     /etc/init.d/tektelic-bstn restart
   - Now your Gateway should be able to connect to the server.
   - You can find the packet forwarder log file in /var/log/pkt_fwd.log
   - You can find the Basic Station log file in /var/log/syslog (If your GW has 3.0.x, 3.1.x, 4.0.x and 4.1.x BSPs)
   - You can find the Basic Station log file in /var/log/bstn.log (If your GW has 3.2.x and 4.2.x or later)

If you have any questions or issues, please reach out to us on our support portal (sign up required - https://support.tektelic.com/portal/en/signin), or support email – support@tektelic.com

6 Verifying Operation – a "Hello World" example

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

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



6.1 Create lambda function for destination rule

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

- Go to AWS IoT Core (https://console.aws.amazon.com/iot/home)
- In the navigation tab, select **Settings** and note ***Endpoint URL***. Leave this tab open as we'll need this further.

- In a new tab, go to the AWS Lambda console (https://console.aws.amazon.com/lambda).
- Click on **Functions** in the navigation pane
- Click on **Create function**
- Select **Author from scratch.** Under Basic information, enter the function name "*sailboatdecoder*" and choose *Runtime Node.js 12.x*. from the drop-down under **Runtime**.
- Click on **Create function**.
- Navigate to *https://tek-aws-gsg.s3.amazonaws.com/lambda.txt* and copy the code for the lambda function.
- Under **Function code**, paste the copied code into the editor under the **index.js** tab.
- At the top of the index.js text, change *INSERT_ENDPOINT_URL_HERE* with the link from the IoT Core Settings tab.
- Once the code has been pasted, choose "**Deploy**" to deploy the lambda code.
- Click on the **Permissions** tab of the lambda function
- Change the Lambda Role Policy permission
  - Under **Execution role**, click on the hyperlink under **Role name**
  - On the **Permissions** tab, find the policy name and click on it
  - Choose **Edit policy**, and choose the **JSON** tab
  - Append the following to the Statement section of the policy to allow publishing to AWS IoT.

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

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

```
{
    "MessageId": "55d122ab-6355-2233-9874-ff47c5222108",
    "WirelessDeviceId": "65d128ab-90dd-4668-9556-fe47c589610b",
    "PayloadData": "AgAAA2cA3QRoLA==",
    "WirelessMetadata":
    {
        "LoRaWAN":
        {
            "DevEui": "647FDAXXXXXXXXX",
            "FPort": 10,
            "DataRate": 0,
            "Frequency": 904500000,
            "Gateways": [
             {
                    "GatewayEui": "647FDAXXXXXXXXX",
                    "Snr": 12.25,
                    "Rssi": -47
             }
             ],
            "Timestamp": "2020-11-10T20:23:56Z",
        }
    }
```

```
}
```

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

```
{
  "raw": "[02, 00, 00, 03, 67, 00, DD, 04, 68, 2C]",
  "port": "10",
  "light_detected": 0,
  "temperature": 22.1,
  "relative_humidity": 22,
  "deveui": "647FDAXXXXXXXXX"
  "timestamp": "1234567890123"
}
```

6.2 Create the Destination rule

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

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

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

1. Go to the AWS IoT console. In the navigation pane, select **Test**, and choose **MQTT client**.
2. Subscribe to the wildcard topic '#" to receive messages from all topics

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


```
{
  "raw": "[02, 00, 00, 03, 67, 00, DD, 04, 68, 2C]",
  "port": "10",
  "light_detected": 0,
  "temperature": 22.1,
  "relative_humidity": 22,
  "deveui": "647FDAXXXXXXXXX"
  "timestamp": "1234567890123"
}
```


6.3 Configuring SNS

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

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

6.3.1 Add a rule for SNS notification

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

- Navigate to the AWS IoT Core console.
- In the navigation pane, choose **Act**. Then, choose **Rules**.
- On the Rules page, choose **Create**
- Enter the **Name** as *text_alert*, and provide an appropriate **Description**
- Under **Rule query statement**, enter the following query:

```
SELECT deveui, "Temperature exceeded 22" as message, relative_humidity,
temperature, time FROM 'project/sensor/decoded' where temperature > 22
```

- Choose **Add action**
- Choose **Send a message as an SNS push notification**
- Choose **Configure action**
- Under **SNS target**, select *text_topic* from the drop-down
- Select *RAW* under **Message format**
- Under **Choose or create a role to grant AWS IoT access to perform this action**, choose **Create role.**
- Enter a name for the role and choose **Add action**
- Choose **Create rule**. You should see a "Success" message, indicating that the rule has been created.

6.4 IoT Analytics

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

6.4.1 Create an IoT Analytics Rule

First create a rule

- Navigate to the AWS IoT Core console.
- In the navigation pane, choose **Act**. Then, choose **Rules**.
- On the Rules page, choose **Create**
- Enter the **Name** as *Visualize*, and provide an appropriate **Description**
- Under **Rule query statement**, enter the following query:

```
SELECT * FROM 'project/sensor/decoded'
```

- Choose **Add action**
- Select **Send a message to IoT Analytics**
- Choose **Configure Action**
- Choose **Quick Create IoT Analytics Resources**
- Under **Resource Prefix**, enter an appropriate prefix for your resources, such as *LoRa*
- Choose **Quick Create**
- Once the **Quick Create Finished** message is displayed, choose **Add action**.
- Choose **Create rule**. You should see a Success message, indicating that the rule has been created.

6.4.2 Configure IoT Analytics

Set up IoT Analytics as follows:

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

```
select temperature, relative_humidity, deveui, time from LoRa_datastore
```

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

6.4.3 Configure QuickSight

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

- Go to AWS Management console.
- From the management console, enter "QuickSight" in the "*Search for services, features..*" search box.

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

6.5 Testing your "Hello World" Application

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

You can also visualize the data set as follows:

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

7 Support

If at any step you encounter problems – feel free to reach out to us on our support portal (sign up required - https://support.tektelic.com/portal/en/signin), or support email – support@tektelic.com