Browan Communications Inc.
No.15-1, Zhonghua Rd., Hsinchu Industrial Park,
Hukou, Hsinchu, Taiwan, R.O.C. 30352
Tel: +886-3-6006899
Fax: +886-3-5972970

Outdoor Micro Gateway

Getting Started Guide

with Basic Station & AWS IoT Core for LoRaWAN

Copyright 2020 Browan Co., Ltd.

Browan Highly Confidential - Controlled Access

2020 © Browan Communications Inc., All Rights Reserved. This document is for planning purposes only, and is not intended to modify or supplement any specifications or warranties relating to products of Browan Communications. Browan may make changes to specifications and descriptions at any time, without notice.

Table of Contents

1 Document Information 3

1.1 Naming Conventions 3

1.2 Revision History (Version, Date, Description of change) 3

2 Overview 3

3 Hardware Description 3

3.1 Product Information 3

4 Setup your AWS account and Permissions 3

4.1 Overview 3

4.2 Set up Roles and Policies in IAM 3

4.2.1 Add an IAM Role for CUPS server 3

4.2.2 Add IAM role for Destination to AWS IoT Core for LoRaWAN 5

4.3 Add the Gateway to AWS IoT 6

4.3.1 Preparation 6

4.3.2 Add the LoRaWAN Gateway 6

4.4 Add a LoRaWAN Device to AWS IoT 6

4.4.1 Preparation 6

4.4.2 Register the Device 7

4.4.3 Verify Profiles 7

4.4.4 Set up a Destination for device traffic 8

4.5 Create the AWS IoT things, certificate and private key 8

5 Set up the Gateway 9

5.1 Set up Gateway hardware 9

5.1.1 Setup of ODU 9

5.1.2 Configuration of ODU 9

5.2 Set up Gateway Software 11

5.2.1 Reset to default 11

5.3 Additional Software References 12

5.4 Configure the Gateway device 12

6 Verifying Operation – a "Hello World" example 13

7 Debugging 14

8 OTA firmware update 15

8.1.1 Custom MQTT message publish 16

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)

1.0 12-Dec-2020 Initial Version

2 Overview

The purpose of this Getting Started Guide is to provide guidelines for AWS IoT Core for LoRaWAN with the outdoor LoRa gateway ODU_BS_AWS (ODU for short)

3 Hardware Description

3.1 Product Information

See https://www.browan.com/product/outdoor-micro-gateway/detail. Additional information is available in the Download links on this page.

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 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.*

- 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. 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**.

Create the corresponding policy

- 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**.

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.
- Minimum software versions 3.00.19, including Basics Station 2.0.5

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.
- Select **Wireless connectivity** in the navigation panel on the left.
- Choose **Intro**, and then choose **Get started**. This step is needed to pre-populate the default profiles.
- Under **Add LoRaWAN gateways and wireless devices**, choose **Add gateway**.
- In the **Add gateway** section, fill in the **GatewayEUI** and **Frequency band (RF Region)** fields.
- Enter a descriptive name in the **Name – optional** field. We do not recommend you leave it blank.
- 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).
- In the section **Provisioning credentials**, choose **Download server trust certificates** to download the CUPS (cups.trust) and LNS (lns.trust) server trust certificates.
- Copy the CUPS and LNS endpoints and save them for use while configuring the gateway.
- Choose **Submit** to add the gateway.

4.4 Add a LoRaWAN Device to AWS IoT

You can skip this section if you don't have devices to add at this time.

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, NwkSkey, FNwkSIntKey, SNwkSIntKey, AppSKey

4.4.2 Register the Device

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**.
- Choose **Continue**
- Under **LoRaWAN specification and wireless device configuration**, enter the **DevEUI** and confirm it in the **Confirm DevEUI** field.
- Enter the remaining fields as per the OTAA/ABP choice you made above.
- Enter a name for your device in the **Wireless device name – optional** field.
- In the **Profiles** section, under **Wireless device profile**, find a drop-down option that corresponds to your device and region.
    - NOTE: Compare your device details to ensure the device profile is correct. If there are no valid default options, you will have to create a new profile (see the section Verify Profiles).
- Choose **Next**
- Choose the destination you created earlier (*ProcessLoRa*) from the drop-down under **Choose destination.**
- Choose **Add device**
- You will see a message saying "*Wireless device added*", indicating that your device has been set up successfully.

4.4.3 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.

4.4.4 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.**
- 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.

4.5 Create the AWS IoT things, certificate and private key

If you plan to use the OTA update feature or AWS IoT Core features, proceed to do these steps.
Create a thing in the AWS IoT registry to represent the ODU:

- In the AWS IoT console, in the navigation pane, choose **Manage**, and then choose **Things**.
- If a **You don't have any things yet** dialog box is displayed, choose **Register a thing**. Otherwise, choose **Create**.
- On the **Creating AWS IoT things** page, choose **Create a single thing.**
- On the **Add your device to the device registry** page, enter a name for your IoT thing and then choose Next. You can't change the name of a thing after you create it. To change a thing's name, you must create a new thing, give it the new name, and then delete the old thing.

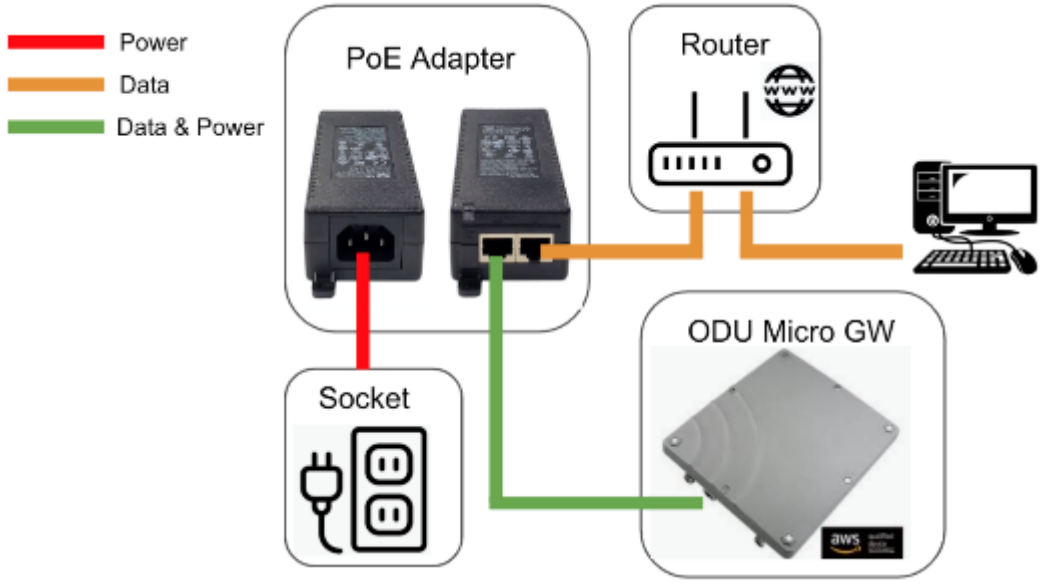
- On the **Add a certificate for your thing** page, choose **Create certificate**.
- Choose the **Download** links to download the certificate, private key, and root CA certificate.
    - Important This is the only time you can download your certificate and private key.
- Choose **Activate**.
- Choose **Attach a policy**.
- For **Add a policy for your thing**, and then choose **Register Thing**.

5 Set up the Gateway

5.1 Set up Gateway hardware

5.1.1 Setup of ODU

ODU is a power over ethernet (PoE) device. First, using an ethernet cable, connect the PoE Adapter's DATA IN port and router's LAN port; then use another cable to connect ODU WAN port and adapter's DATA&POWER OUT port to turn on the ODU, as shown in the illustration below.

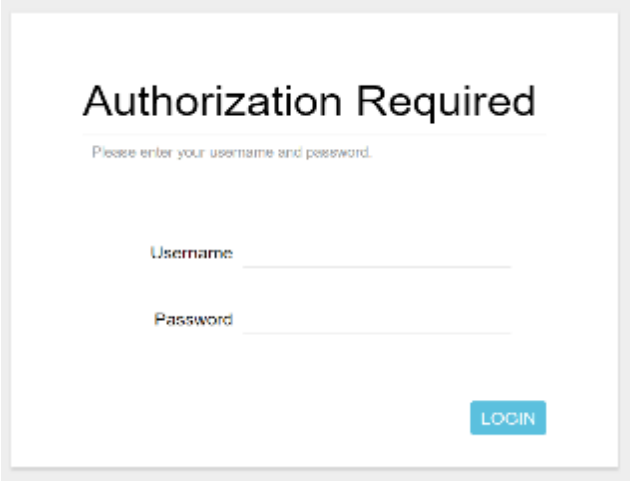


5.1.2 Configuration of ODU

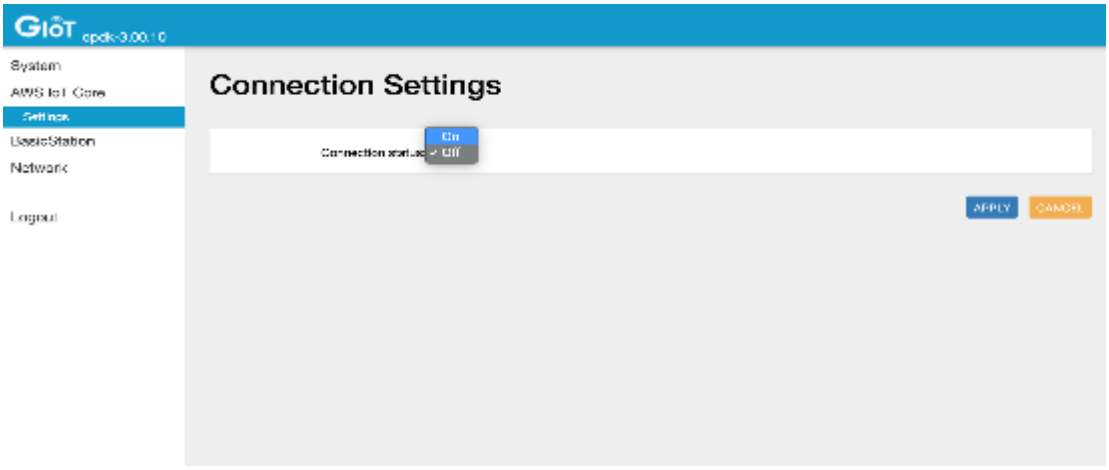
ODU uses DHCP, which assigns dynamic IP addresses as a default. Be sure to connect ODU to the router through PoE adapter, as shown in the above illustration. When the ODU turns on, the router will automatically allocate an unused IP address to the ODU. You must specify the IP address <ODU_DHCP_IP> which can be obtained from the router web-based setup page.

Open the ODU WEB GUI page via the <ODU_DHCP_IP> in your browser then do login

http://<ODU_DHCP_IP>/cgi-bin/login.cgi Username: **admin** Password: **admin**



Navigate to the AWS IoT Core → Settings page, change the Connection Status to On.



Input the following configuration then choose APPLY.

Connection status: On
Endpoint Address: Please use the AWS IoT console. In the left panel choose Settings. The endpoint is listed under Custom endpoint
Endpoint Port: 443 or 8883 (Default: 443)
Thing Name: Created thing name from AWS IoT Core
Root CA: AWS Root CA Download the file at AWS Root CA and then upload here
Client Certificate: Upload the Thing's Certificate. Please refer to **4.5 Create the AWS IoT things, certificate and private key**"
Client Private Key: Upload the Thing's Private Key. Please refer to **4.5 Create the AWS IoT things, certificate and private key**"
MQTT Keepalive Interval: 30

NOTE:
Gateway will publish the keep-alive message with the AWS IoT demo topic: iotdemo/browan-<ethernet_mac>/keepalive

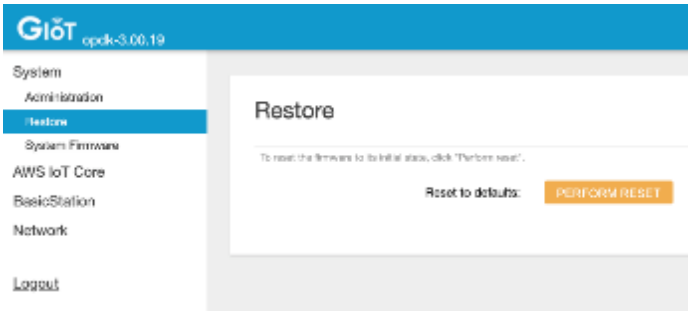
Example:
Ethernet MAC: 1C:49:7B:A9:EE:1C
iot demo topic: iotdemo/browan-1c497ba9ee1c/keepalive

5.2 Set up Gateway Software

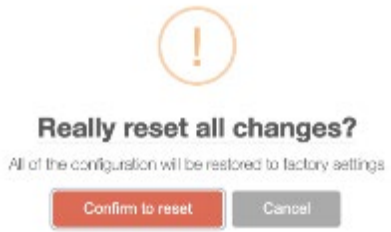
5.2.1 Reset to default

To restore all of the configuration to factory settings on the gateway, you need:

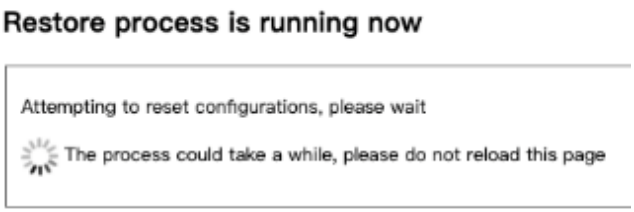
- Go to **System** → Restore, click **PERFORM RESET**



- Click **Confirm to reset**



- Within a few minutes, all of the configuration will be restored to factory settings.

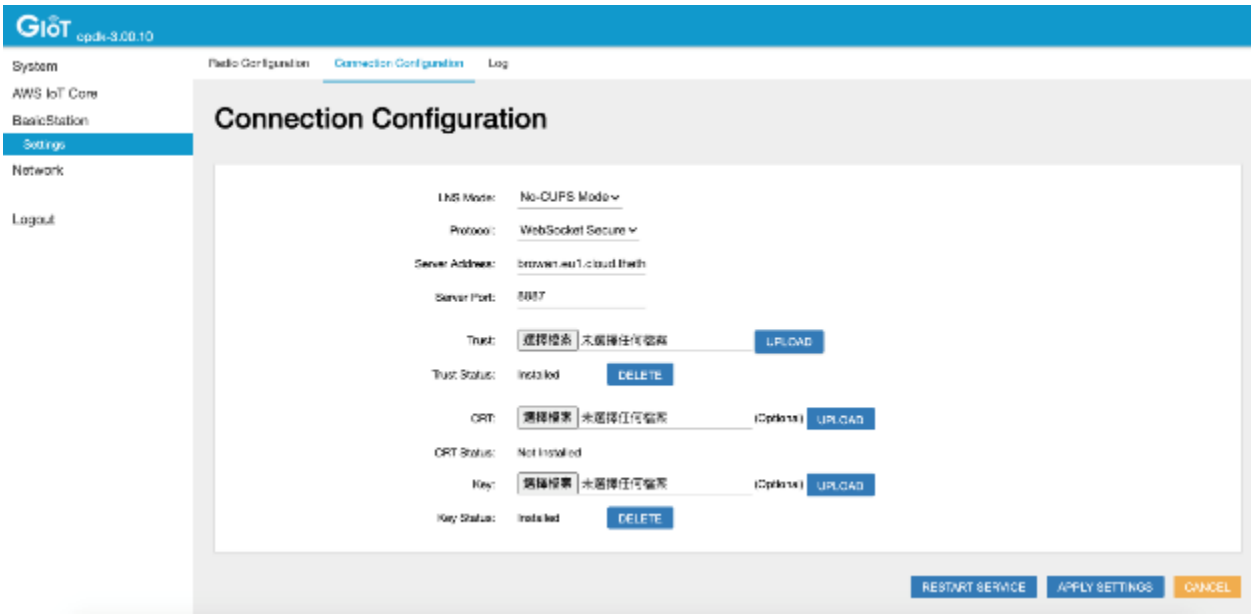


5.3 Additional Software References

https://www.browan.com/news/Rd/detail

5.4 Configure the Gateway device

Go to the **BasicStation** → **Settings** page, click **Connection Configuration**.



Input the following configuration then choose **APPLY SETTINGS**

LNS Mode: **Normal mode** or **No-CUPS Mode**. **Normal mode** is CUPS mode base; **No-CUPS Mode** is directly connecting to LNS.
Protocol: **HTTP** or **HTTPS** for Normal mode; **WebSocket** or **WebSocket Secure** for No-CUPS Mode.
Server port:
Trust: Upload the Trust file. It's a server's CA certificate which enables the Station to establish trust with the server. If the URI indicates a TLS-based scheme, this file must be present and contain a PEM-encoded X509 certificate.
Trust Status: If the trust file is uploaded, it will show **Installed**. Otherwise shows **Not Installed**.
CRT: Upload the CRT file. It's a Station's own certificate, if TLS client authentication is being used. If this is not empty, the corresponding key file must contain the private key matching this certificate.
CRT Status: If the CRT file is uploaded, it will show **Installed**. Otherwise shows **Not Installed**.
Key: Upload the Key file. It's a Station's private key matching the corresponding certificate (.crt) file. If the certificate file is missing or empty, this file, if present, shall contain an authorization token submitted in the HTTP header field *Authorization when making requests to the server. Choose
Key Status: If the Key file is uploaded, it will show **Installed**. Otherwise shows **Not Installed**.

6 Verifying Operation – a “Hello World” example

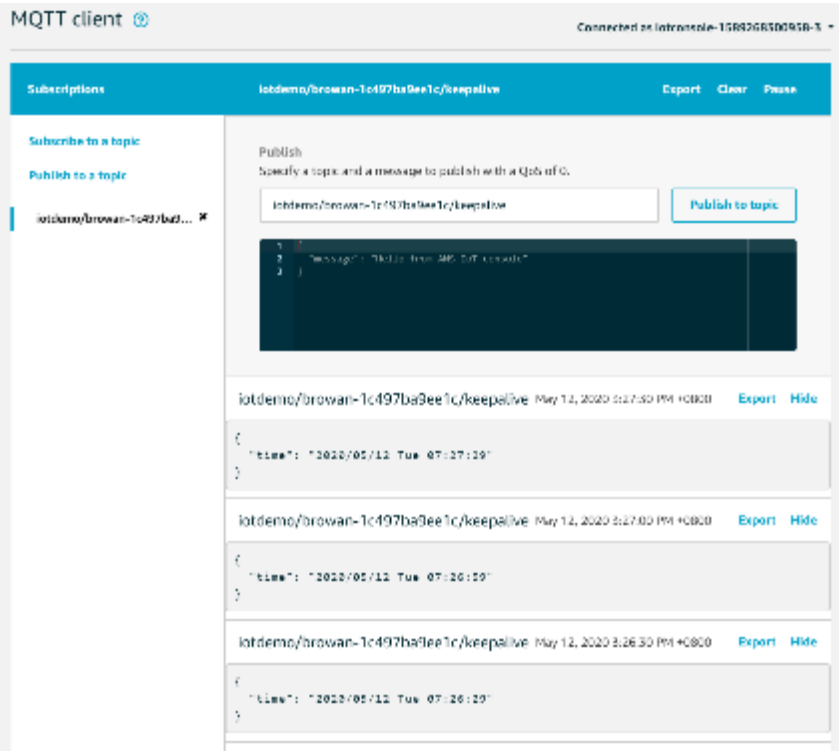
Once setup is completed, provisioned devices can join the network and start to send messages. Messages from devices can then be received by AWS IoT Core for LoRaWAN and forwarded to the IoT Rules Engine.

The “Hello World” example shown here is to demonstrate that traffic from the gateway can be viewed on the AWS IoT console.

To view this traffic:

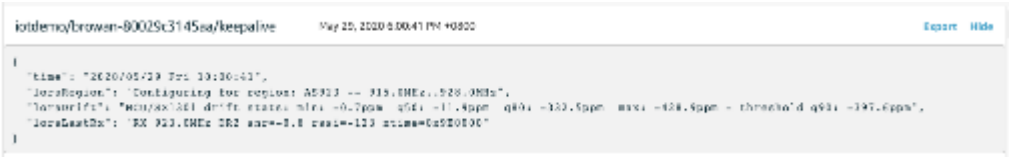
- Go to the AWS IoT Core console and choose Test from the navigation panel.
- Select the MQTT Client
- Subscribe to the AWS IoT demo topic iotdemo/browan-1c497ba9ee1c/keepalive as shown below.

AWS IoT Core will receive the keep-alive json data every 30 seconds.



LoRa fields

If the ODU received the latest up-link from the LoRa sensor



7 Debugging

By using SSH, user can remotely log in to ODU and run some commands to get different log levels

- Login to ODU through **ssh**
  - Account: root
  - Password: browan@[The latest 3 bytes of the ODU's base MAC address in capital letters]

  e.g.

  The ODU's base MAC on the back label is 80029C1A2B3C
  Login password is **browan@1A2B3C**

  - Command: ssh [account]@[ODU DHCP IP address]

  e.g.

  BrowanMBP:~ browan_mbp$ ssh root@192.168.88.136
  root@192.168.88.136's password: ***browan@1A2B3C***

  NOTE: It is recommended that this password be changed to a more secure one.

- Trace basic-station log with **tail** command
  - Command: tail -f /tmp/station.log

- Change basic-station debug levels with **vim** command then modify the field **log_level**
  - Command: vim /opt/basicstation/station.conf

```
"station_conf": {
    "log_file":  "/tmp/station.log",
    "log_level": "DEBUG",  /* XDEBUG,DEBUG,VERBOSE,INFO,NOTICE,WARNING,ERROR,CRITICAL */
    "log_size":  10000000,
```

```
    "log_rotate":  3,
     "gps": "/dev/ttyS3",
     "pps": "gps"
}
```

- Restart basic-station through init.d service command to get changed log levels.
    - Command: /etc/init.d/station.service restart

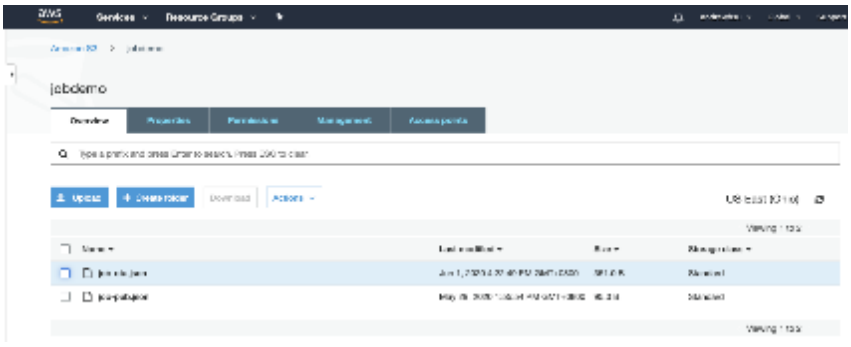
8 OTA firmware update

The Browan Firmware Release Note describes what's new and changed in AWS ODU FW RELEASE. You can use OTA updates to install the latest version of the ODU firmware. We recommend that you check the firmware release before you use OTA updates.

❇ Note: Please DON'T modify the OTA-job json file.

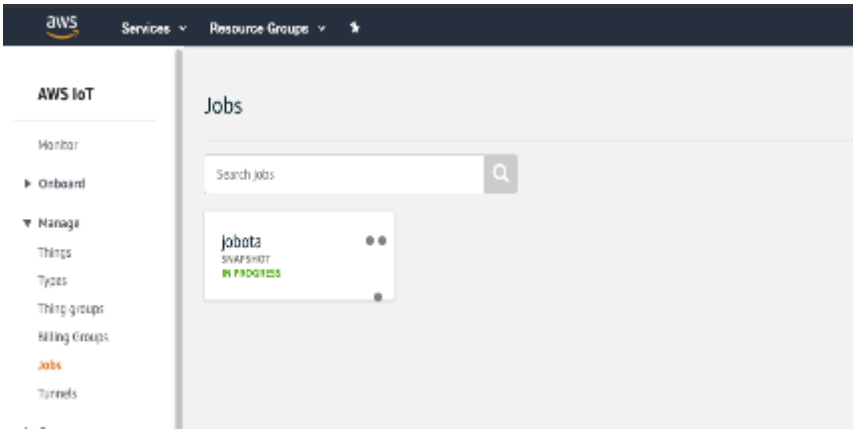
- Check the ODU FW version by Web GUI page before OTA

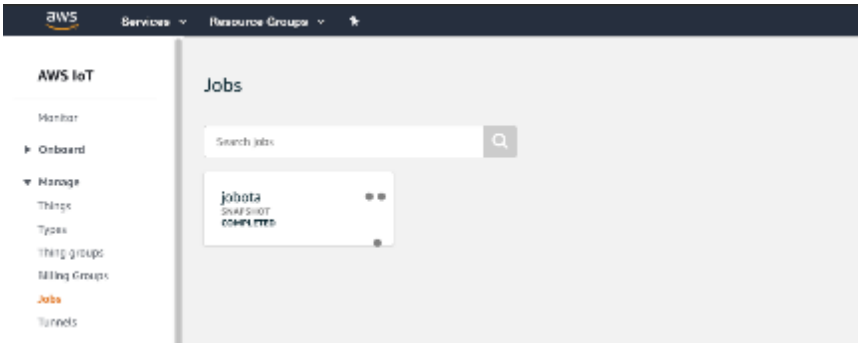
- Create a bucket from Amazon S3 so you can store job documents and upload the ***job-ota.json*** from AWS IoT Core Job here: Creating a bucket - Amazon Simple Storage Service



- Navigate to the AWS IoT Core console. In the navigation pane, choose **Manage,** and then **Jobs**
- Click **Create** to create a job

- Click **Create custom job**
- Enter the **Job ID** *jobota* on Create a job page
- Click **Select** to select a device in Things list at **Select devices to update** on Create a job page
- Click **Select** to select a job file job-ota.json in S3 storage at **Add a job file** on Create a job page
- Click **Next** to completed
- Wait a few minutes; the created job *jobota* status will be changed to **COMPLETED** from IN **PROGRESS**





- ODU FW version will be updated to the stable version

8.1.1 Custom MQTT message publish

You can publish a specific MQTT message by AWS IoT Jobs. Job is a remote operation that instructs ODU to publish messages to AWS IoT. To create a job, you must create a job document that is a description of the remote operations to be performed. Here is a job document jobDoc_forPub, save it to ***job-pub.json***.

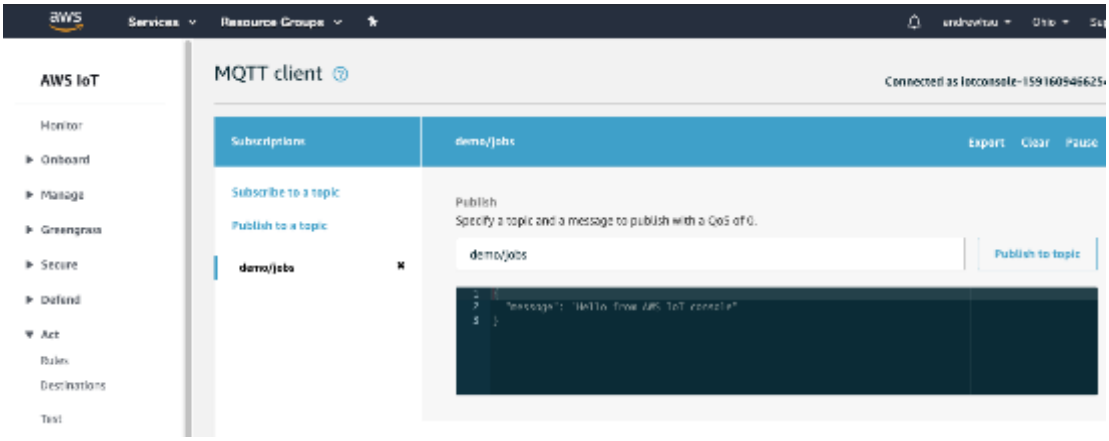
- The job document should contain the following information.

```
{
  "action" : "publish",
```

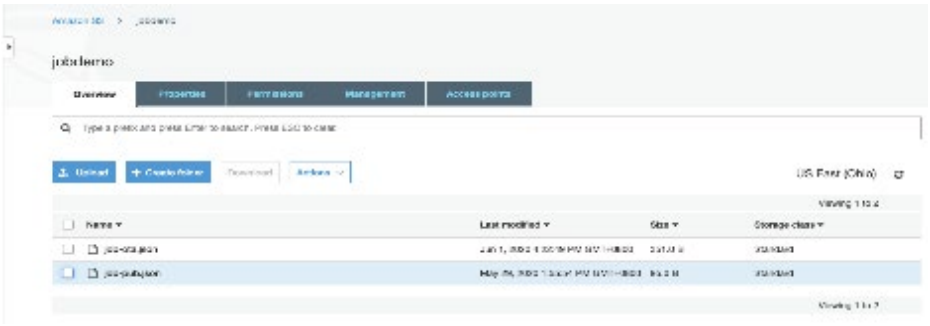
```
  "topic" : "demo/jobs",
  "message" : "awsiot job-pub demo"
}
```

action: Must **NOT** be modified.
topic: Configure what you want to subscribe from the AWS IoT console.
message: Configure what you want to receive from the **topic**.

- Navigate to the AWS IoT Core console , Choose **Act** → **Test**, subscribe the **topic** on MQTT client page.

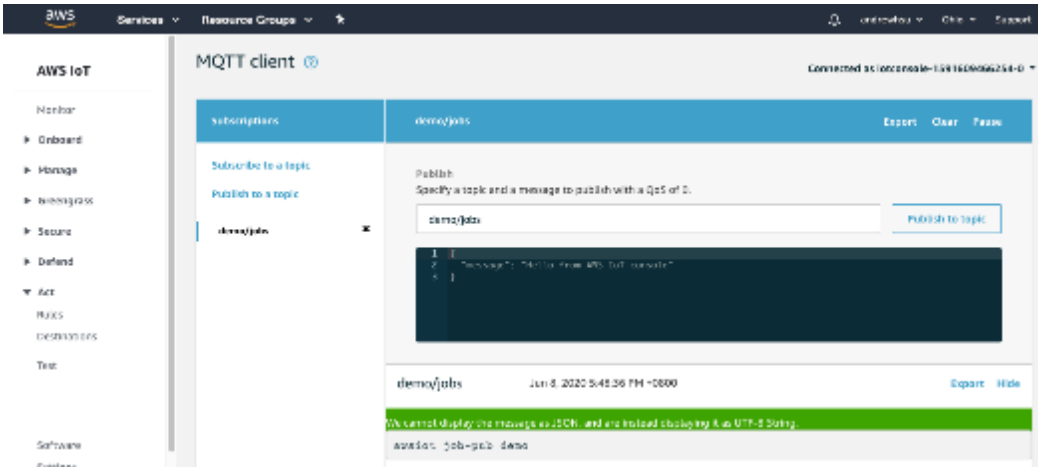


- Create a bucket from Amazon S3 so you can store job documents and upload the job file ***job-pub.json*** from AWS IoT Core Job.



- Click **Create** to create a job
- Click **Create custom job**
- Enter the **Job ID** *jobput* on Create a job page
- Click **Select** to select a device in Things list at **Select devices to update** on Create a job page
- Click **Select** to select a job file in S3 storage at **Add a job file** on Create a job page

- Within a few minutes, AWS IoT Core console will receive a message on MQTT client page



- The created job *jobpub* status will be changed to **COMPLETED** from IN **PROGRESS** on Jobs page.