



LogBox Wi-Fi

AWS VIA MQTT CONFIGURATION MANUAL



Applies to devices with firmware version starting with V1.1x.

1. PRESENTATION

Through the MQTT protocol, the **LogBox Wi-Fi** can be configured to communicate with Amazon Web Services (AWS). To do this, follow the steps in this manual.

2. CONFIGURING AWS

To perform the communication and pairing between the **LogBox Wi-Fi** and AWS, you need the **NXperience** software, available on our website, and optionally the **MQTTBox**.

After that, the following procedures must be performed:

1. Create an account in Amazon Web Services.
2. Once the account is created, access the **Services** tab in the top menu and, in the **Internet of Things** subsection, select the **IoT Core** option.

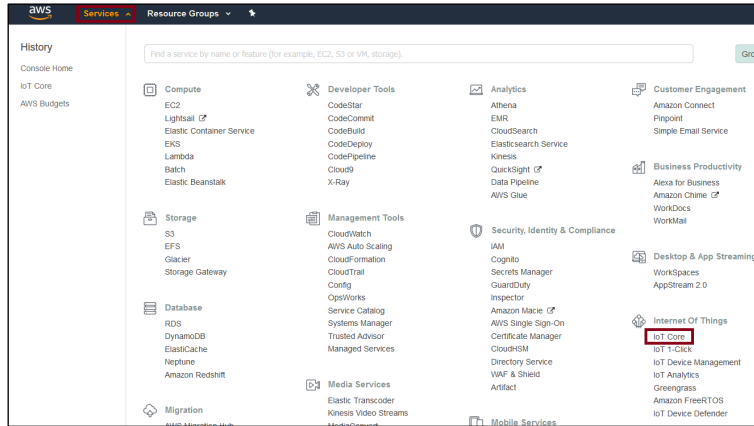


Fig. 01 – AWS Services

3. In the **IoT Core** service, open the **Secure** tab in the left menu and select the **Policies** option. Once the option has been selected, click on **Create a policy**.

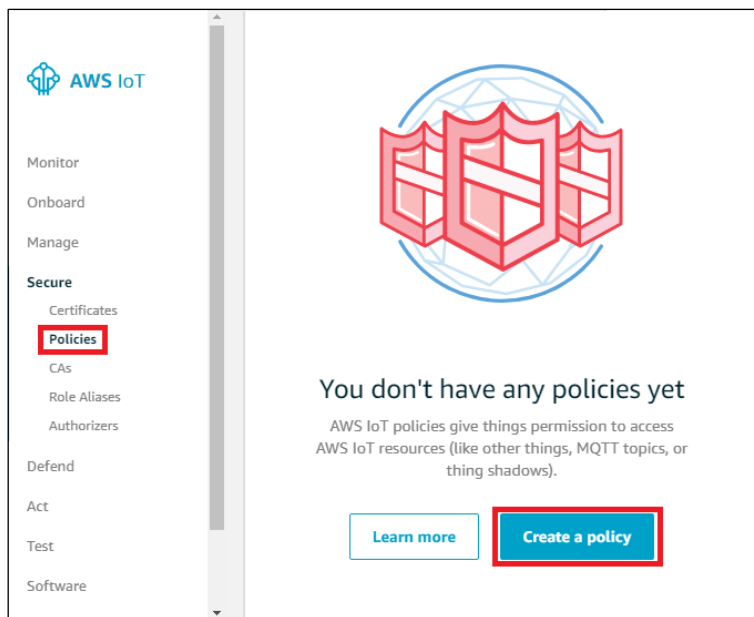


Fig. 02 – Create a policy

- In the **Name** field, name the policy to be created. In the **Action** field, type: "iot:*". Edit the **Resource ARN** field for "*". Check the **Allow** field and click the **Create** button.

Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the [AWS IoT Policies documentation page](#).

Name
FirstPolicy

Add statements
Policy statements define the types of actions that can be performed by a resource. Advanced mode

Action
iot:*

Resource ARN
*

Effect
 Allow Deny Remove

Add statement

Create

Fig. 03 – New policies configuration

- After creating a new policy, still in the **Secure** tab, in the left menu of the **IoT Core** service, select the **Certificates** option. Once this option has been selected, click **Create a certificate**.

AWS IoT

Monitor
Onboard
Manage
Secure
Certificates
Policies
CAs
Role Aliases
Authorizers
Defend
Act
Test
Software

You don't have any certificates yet
Certificates help things establish a secure connection.

Learn more Create a certificate

Fig. 04 – Create a certificate

- Click **Create certificate** in the **One-click certificate creation** tab.

One-click certificate creation (recommended)
This will generate a certificate, public key, and private key using AWS IoT's certificate authority.

Create certificate

Fig. 05 – One-click certificate creation

7. Download the **A certificate for this thing** and the **Private Key**.

In order to connect a device, you need to download the following:		
A certificate for this thing	c98c6ffc20.cert.pem	Download
A public key	c98c6ffc20.public.key	Download
A private key	c98c6ffc20.private.key	Download

Fig. 06 – Downloads

8. Download the certificate from a root CA.

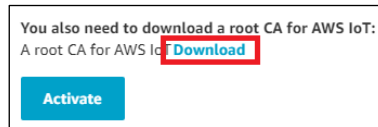


Fig. 07 – root CA

9. In this new window, click the **Amazon Root CA 1** link to download AWS Root CA certificate.

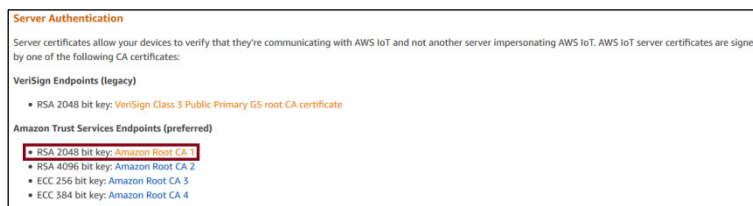


Fig. 08 – Root CA 1

10. When you click on the **Amazon Root CA1** link, the certificate will be displayed on the webpage in text format. To save it, just click on **Ctrl + S**.

For the TLS v1.2 security layer, you must have the 3 certificates, similar to the ones below:

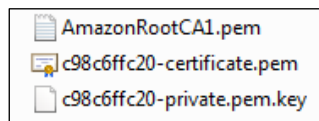


Fig. 09 – Examples

11. Before completing the procedure, click **Attach a policy** and append the policy created in **Step 4**.



Fig. 10 – Attach a policy

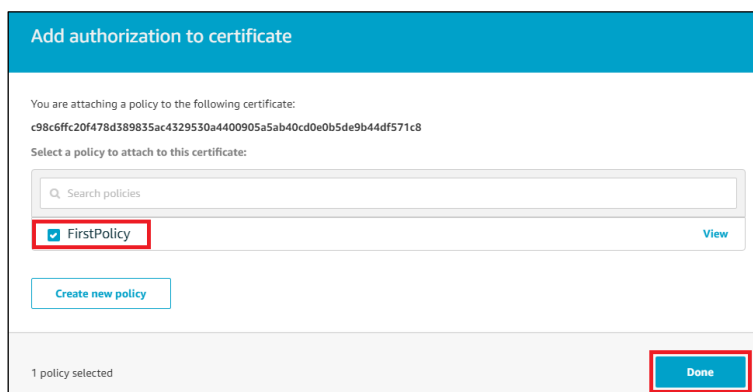


Fig. 11 – Select a policy

12. Check the created certificate, access the list of **Actions** and click **Activate**.

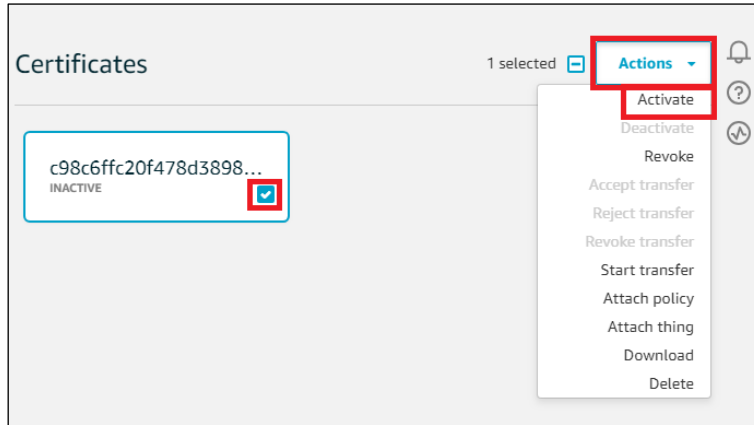


Fig. 12 – Activate

13. On the side menu, expand the **Manage** tab and select the **Things** option. In the next window, click **Register a thing**.

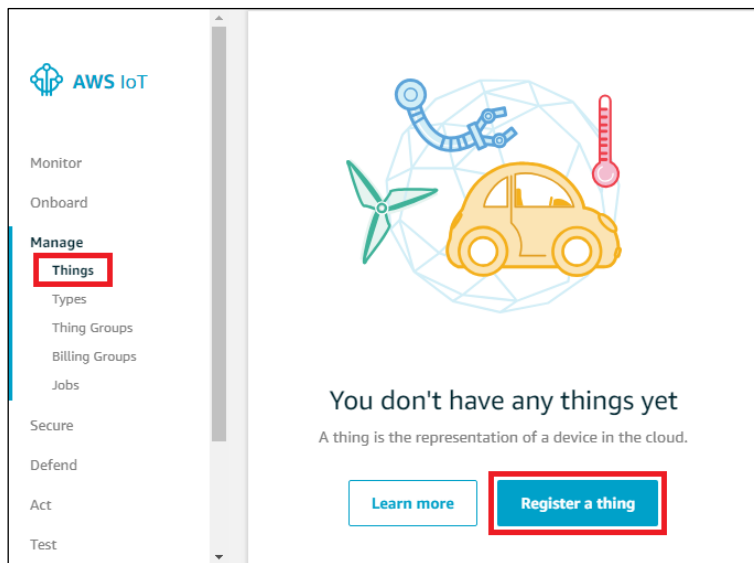


Fig. 13 – Register a thing

14. In the new tab, click **Create a single thing**.



Fig. 14 – Create a single thing

15. Name the device in the **Name** field.

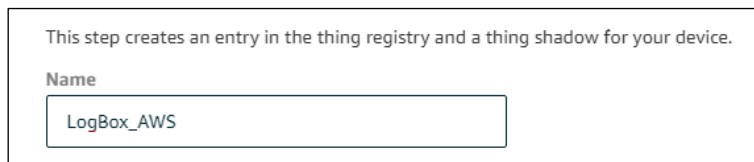


Fig. 15 – Name the device

16. Create a device type by clicking the **Create a type** button.

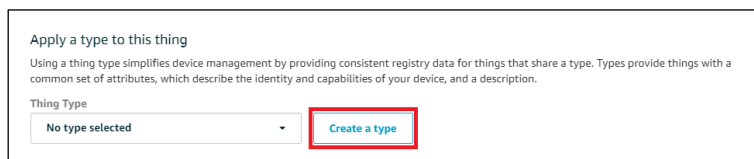


Fig. 16 – Create a type

17. In the new windows, assign a name for the type and click **Create a thing type**.

Create a thing type

This will help you organize, categorize, and search for your things.

Name
Data_Logger

Description
Describe this thing type

Set searchable thing attributes
You can define up to three attributes for a thing type. Things associated with this type can be searched by using these fields.

[Add another](#)

[Cancel](#) [Create thing type](#)

Fig. 17 – Create thing type

18. After create a type, click **Next**. In the new window, click **Create thing without certificate**.

Skip certificate and create thing
You will need to add a certificate to your thing later before your device can connect to AWS IoT.

[Create thing without certificate](#)

Fig. 18 – Create thing without certificate

19. In the side menu, select the **Secure** tab and access the **Certificates** option. Check the existing certificate, open the **Actions** tab and click **Attach thing**.

Certificates 1 selected

[Actions](#)

- Activate
- Deactivate
- Revoke
- Accept transfer
- Reject transfer
- Revoke transfer
- Start transfer
- Attach policy
- Attach thing**
- Download
- Delete

c98c6ffc20f478d3898...
ACTIVE

Fig. 19 – Attach thing

20. In the new window, check the item to be attached and click **Attach**.

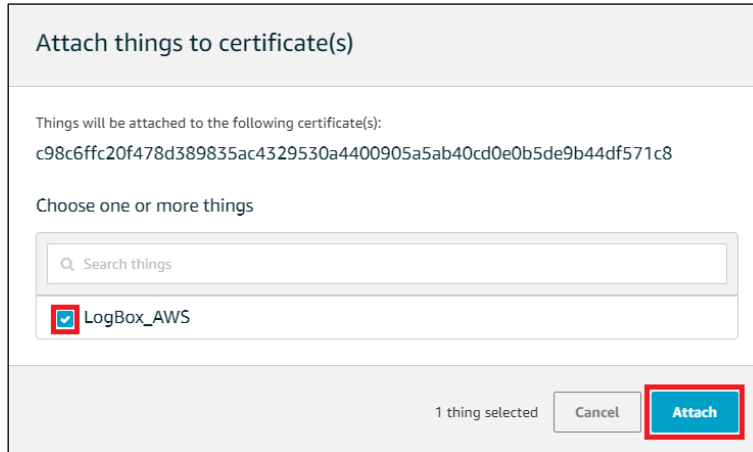


Fig. 20 – Attach things to certificates

21. In the side menu, access the **Settings** option. Save the AWS account custom endpoint.

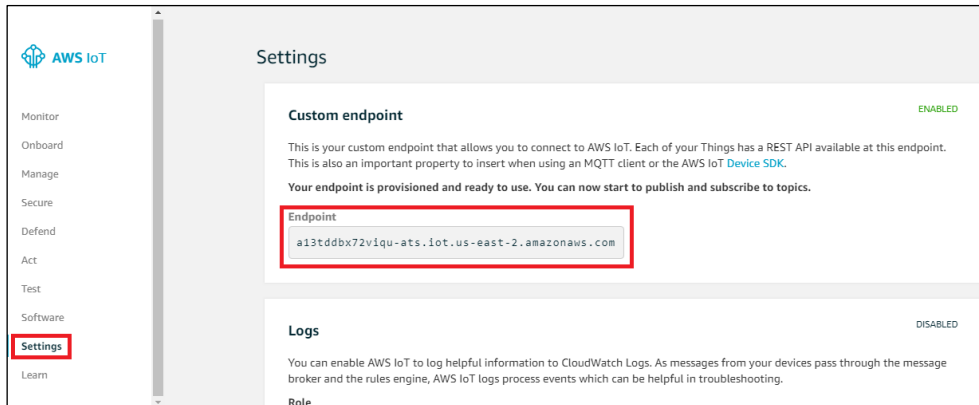


Fig. 21 – Custom endpoint

22. Run **NXperience**. Connect **LogBox Wi-Fi** via USB interface or via Modbus-TCP. Enable the Wi-Fi interface on the **Wi-Fi** tab in the **Communication NXperience** tab. Access the **MQTT** tab and configure it with the following parameters:

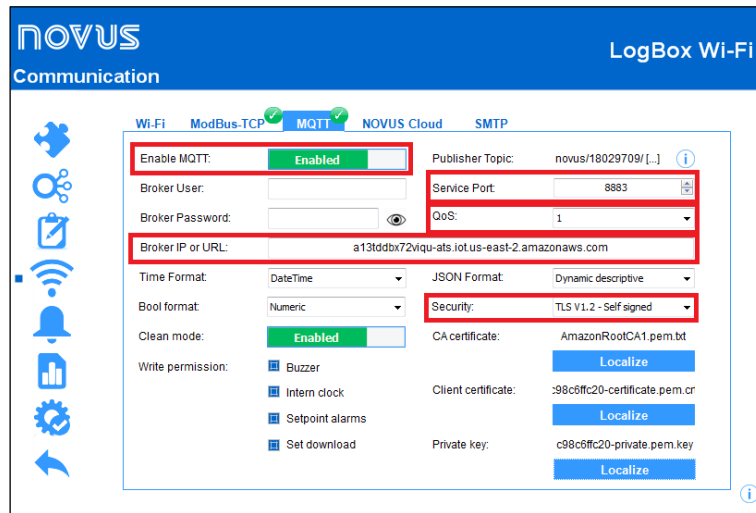


Fig. 22 – Communication

- **Enable MQTT:** Enable the MQTT protocol.
 - **Service Port:** 8883.
 - **QoS:** 1 (AWS does not allow QoS 2).
 - **Broker IP or URL:** Insert Endpoint saved in **Step 21**. **Ex.:** a2jsauz3jc3c1e-ats.iot.us-east-2.amazonaws.com.
 - **Security:** TLS V1.2 – Self signed.
 - **CA Certificate:** Load the Amazon Root CA certificate. **Ex.:** AmazonRootCA1.pem.
 - **Client Certificate:** Load the "thing" certificate. **Ex.:** 993cb74d0b-certificate.pem.crt.txt.
 - **Private Key:** Load the private key certificate. **Ex.:** 993cb74d0b-private.pem.key.
23. Send the new configurations to the **LogBox Wi-Fi**.
24. Return to the AWS console and, in the side menu, select the **Test** option. In the **Subscription topic** field, type "**novus/#**" and click on **Subscribe to topic**.

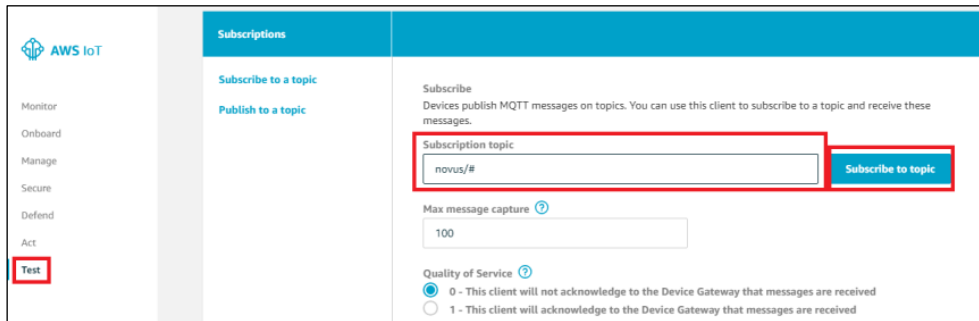


Fig. 23 – Subscription topic

The "**novus/#**" topic will be as shown in **Fig. 24**:

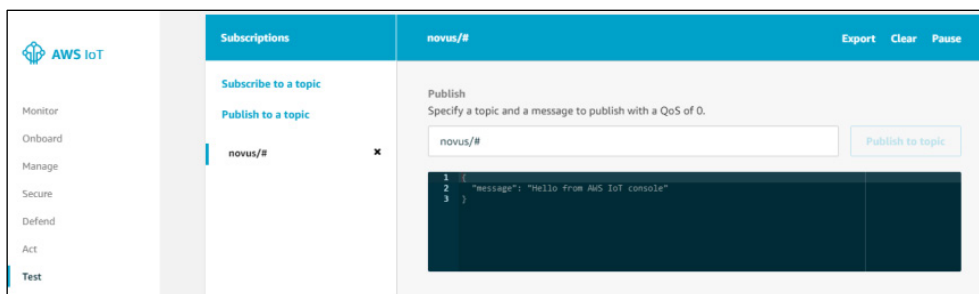


Fig. 24 – "novus/#" topic

25. Restart the device and wait for the previously programmed data to be sent. This procedure will allow you to observe the data being delivered to the Broker, as shown in Fig. 25, Fig. 26 and Fig. 27:

```
novus/18141554/config          30 de out de 2018 12:19:02

{
  "n_channels": 4,
  "timestamp": 43403.59122685,
  "frame_format": "array_static",
  "channels_enabled": [
    0,
    1,
    1,
    0
  ],
  "hash": "9622D6DB0D364A3EA13D17C5D6978746D5A2AFB7",
  "gmt": 0,
  "tag_channels": [
    "",
    "Analog1",
    "Analog2",
    ""
  ],
  "tag_units": [
    "",
    "Celsius",
    "Fahrenheit",
    ""
  ],
  "sp_alarm_low": [
    0,
    0,
    0,
    0
  ],
  "sp_alarm_high": [
    0,
    0,
    0,
    0
  ]
}
```

Fig. 25 – /config topic

```
novus/neighbor                30 de out de 2018 12:19:02

{
  "model": "LogBox Wi-Fi",
  "serial": 18141554,
  "ip": "10.51.11.149",
  "mac": "B0:38:29:7B:D6:C5",
  "lqi": -45,
  "firmware_version": 1.01
}
```

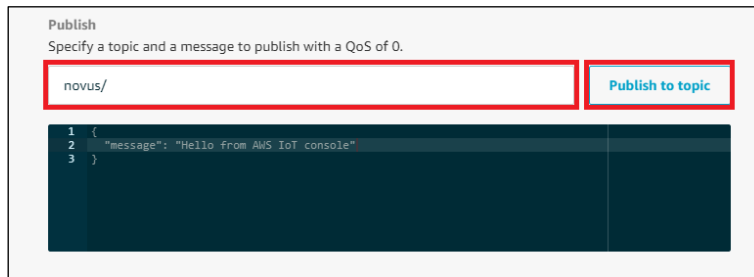
Fig. 26 – /neighbor topic

```
novus/18141554/log/channels   30 de out de 2018 12:19:01

{
  "n_channels": 4,
  "timestamp": 43403.59122685,
  "battery": 0,
  "value_channels": [
    0,
    26,
    1,
    0
  ],
  "alarm_low": [
    0,
    0,
    0,
    0
  ],
  "alarm_high": [
    0,
    0,
    0,
    0
  ],
  "buzzer_state": 0
}
```

Fig. 27 – /log/channels topic

26. In order to perform connectivity tests with the server, you can publish a message in a topic. You must type "novus/" in the blank field in the "novus/#" topic, as shown in Fig. 28, and click **Publish to topic**.



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

novus/ Publish to topic

```
1 {  
2   "message": "Hello from AWS IoT console"  
3 }
```

Fig. 28 – Publish

By default, the message "Hello from AWS IoT console" will be sent:

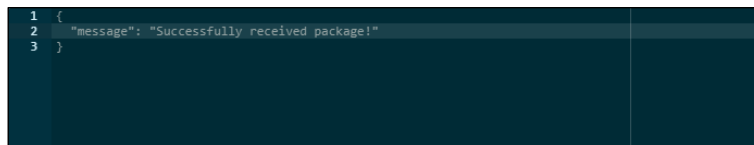


novus/ Nov 28, 2018 1:45:28 PM -0200 Export Hide

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

Fig. 29 – Default message

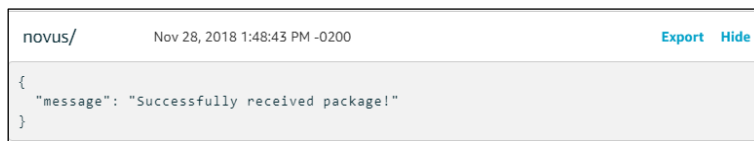
The message can be modified by editing the field represented by Fig. 30:



```
1 {  
2   "message": "Successfully received package!"  
3 }
```

Fig. 30 – Successfully received package

We should obtain the following result:



novus/ Nov 28, 2018 1:48:43 PM -0200 Export Hide

```
{  
  "message": "Successfully received package!"  
}
```

Fig. 31 – Edit the message

3. OPTIONAL STEPS

Steps 24, 25 and 26 can optionally be performed in the MQTT Box software, which can be downloaded for free via the link <http://workswithweb.com/mqttbox.html>.

27. Run the MQTT Box software and click **Create MQTT Client**.

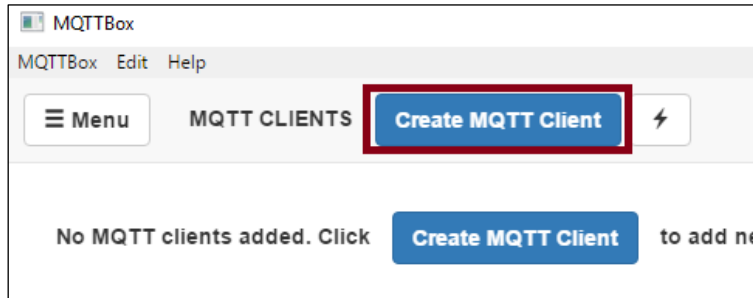


Fig. 32 – Create MQTT Client

28. In the configuration window, fill in the available fields with the following values:

- **MQTT Client Name:** Enter client name. **Ex.:** AWS Broker.
- **Protocol:** mqtt / tls.
- **Host:** Enter the Endpoint address saved in **Step 21**. **Ex.:** a2jsauz3jc3c1e-ats.iot.us-east-2.amazonaws.com.
- **SSL / TLS Version:** TLSv1.2.
- **SSL / TLS Certificate Type:** Self signed certificates.
- **CA File:** Load the Amazon Root CA certificate. **Ex.:** AmazonRootCA1.pem.
- **Client certificate file:** Load the "thing" certificate. **Ex.:** 993cb74d0b-certificate.pem.crt.txt.
- **Client key file:** Load the private key certificate. **Ex.:** 993cb74d0b-private.pem.key.

You do not need to make any changes to the other settings.

The screenshot shows the MQTT Box configuration window. The fields are organized into several sections. The 'MQTT Client Name' field is set to 'AWS Broker'. The 'MQTT Client id' field is set to 'a271f1dbd-3cf0-46f9-9c75-54f2c269fc18'. The 'Append timestamp to MQTT client id?' checkbox is checked. The 'Broker is MQTT v3.1.1 compliant?' checkbox is checked. The 'Clean Session?' checkbox is checked. The 'Auto connect on app launch?' checkbox is checked. The 'Protocol' dropdown is set to 'mqtt / tls'. The 'Host' field is set to 'a2jsauz3jc3c1e-ats.iot.us-east-2.amazonaws.com'. The 'SSL / TLS Version' dropdown is set to 'TLSv1.2'. The 'SSL / TLS Certificate Type' dropdown is set to 'Self signed certificates'. The 'CA file' field is set to 'AmazonRootCA1.pem'. The 'Client certificate file' field is set to '993cb74d0b-certificate.pem.crt.txt'. The 'Client key file' field is set to '993cb74d0b-private.pem.key'. The 'Client key passphrase' field is empty. The 'Queue outgoing QoS zero messages?' checkbox is checked. The 'Username' field is empty. The 'Password' field is empty. The 'Reschedule Pings?' checkbox is checked. The 'KeepAlive (seconds)' field is set to '10'. The 'Reconnect Period (milliseconds)' field is set to '1000'. The 'Connect Timeout (milliseconds)' field is set to '30000'. The 'Will - Retain' checkbox is unchecked. The 'Will - Topic' dropdown is set to 'Will - Topic'. The 'Will - QoS' dropdown is set to '0 - Almost Once'. The 'Will - Payload' field is empty. There are 'Save' and 'Delete' buttons at the bottom.

Fig. 32 – MQTT Box configuration

29. After saving the settings above, check if the MQTT client is connected.

In the **Topic to subscribe** field, type "novus/#". By using this wildcard, the client will sign up for all topics that **NOVUS** devices post.

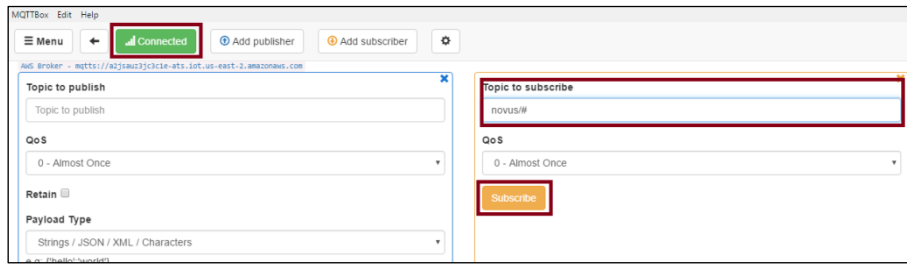


Fig. 33 – Topics subscribe

30. Wait for the device to reconnect to the Wi-Fi network.

Run the **MQTT Box** software and verify that the data is correctly delivered to the Broker, as shown in **Fig. 34**:



Fig. 34 – Broker MQTT