SENSECAP

LoRaWAN Gateway and Wireless Sensor User Guide

How to Work with 3rd-party Standard LoRaWAN Gateway or TTN Server

Version: V1.2

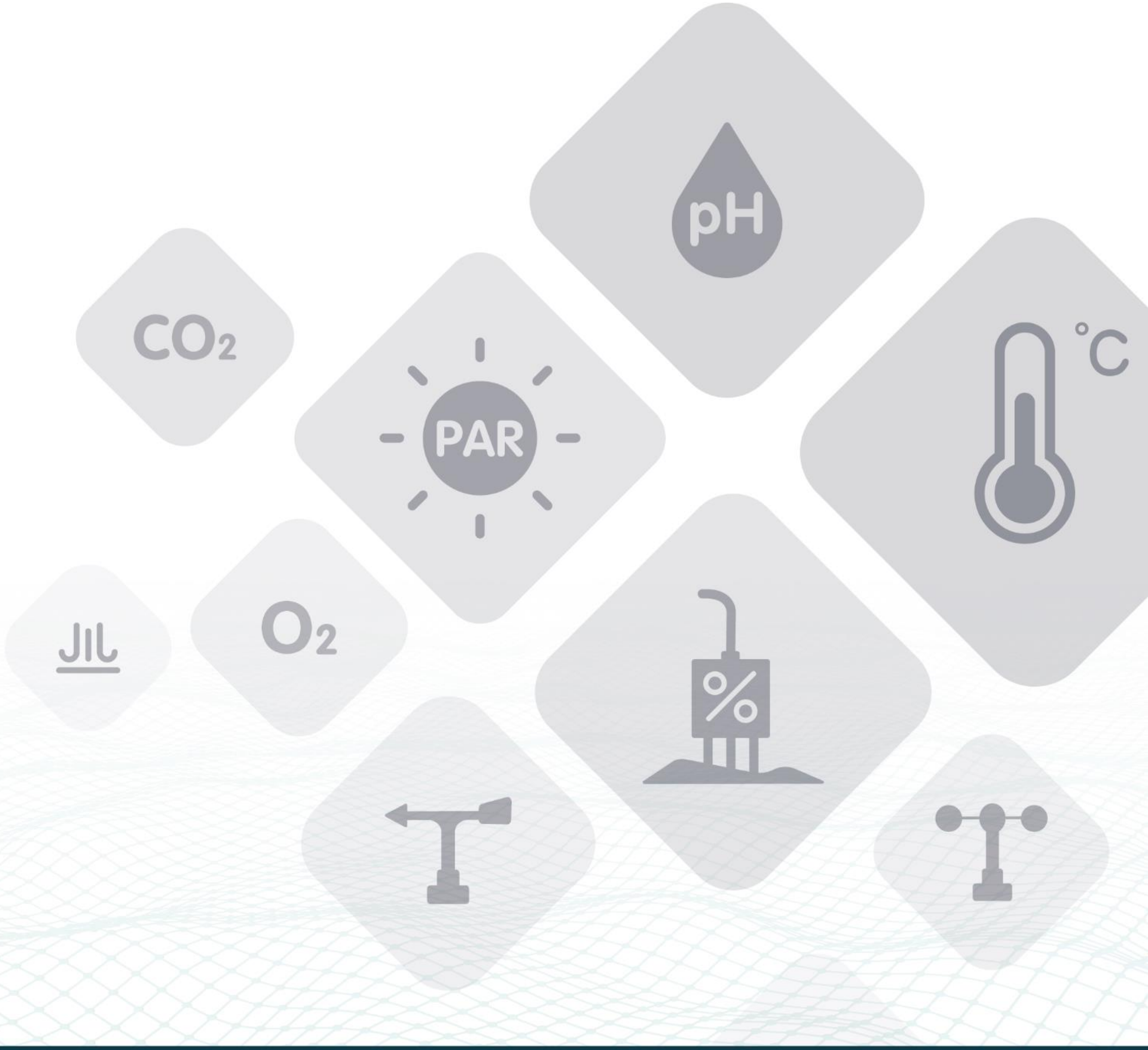


Table of Contents

1 Product Introduction

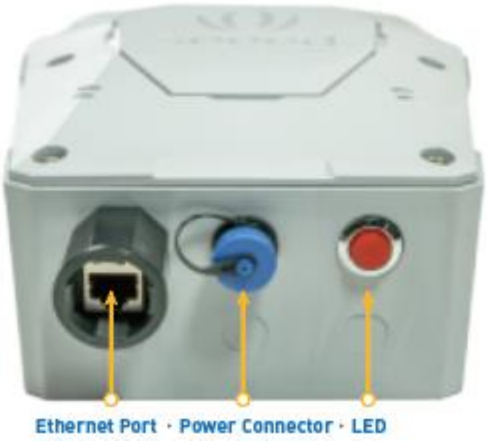
SenseCAP is an industrial wireless sensor network that integrates easy-to-deploy hardware and data API services, enabling low-power, long-distance environmental data collection. SenseCAP includes several versions, such as LoRaWAN, LoRaPP, etc.

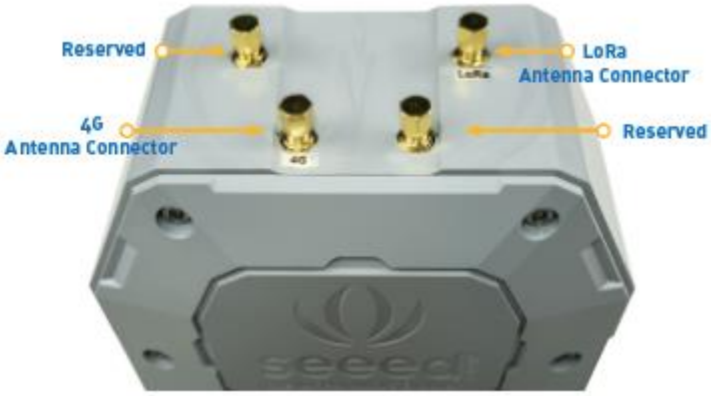
SenseCAP LoRaWAN version products include LoRaWAN Gateways and Sensor Nodes. Based on the LoRaWAN protocol, it can realize one-to-many, long-distance networking and bilateral communication. The LoRaWAN Gateway supports Ethernet and 4G. The Sensor Node is powered by a high-capacity battery that lasts up to 3 years (if uploading data once every hour). It also supports hot-swap, making it easy for maintenance and upgrading.

Main Features:

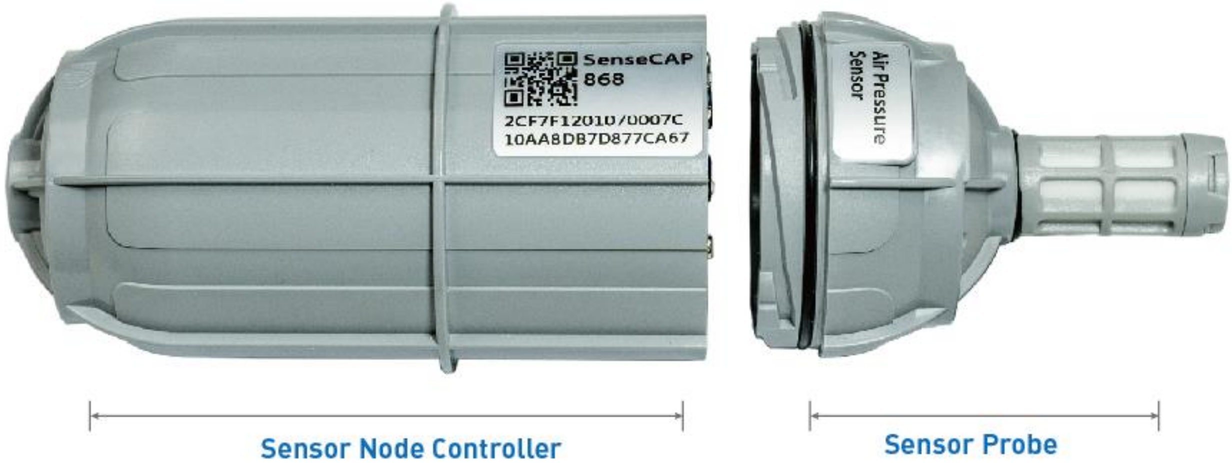
- Gateway: High-performance Cortex A8 1GHz processor
- Gateway uses multiple methods to connect to the Internet: 4G and Ethernet
- Gateway supports third-party TTN account and server
- Sensors support LoRaWAN v1.0.2 protocol and are suitable for standard LoRaWAN Gateway
- Super long-distance communication: 10km in the line-of-sight scenario, 2km in the urban scenario
- Industrial protection rating IP66-rated enclosure, suitable for the outdoor environment at -40℃~70℃
- Easy-to-deploy, enabling people without engineering background to install the devices quickly

LoRaWAN Gateway:





LoRaWAN Sensor Node:

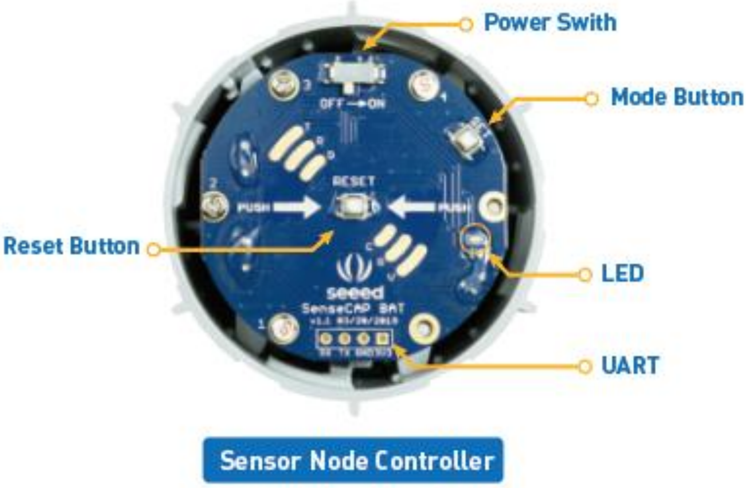


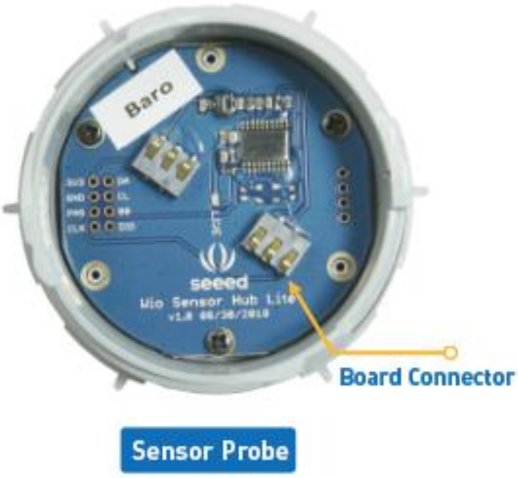
Sensor Node Controller

- LoRa Communication module
- Ultra-low power microcontroller
- Battery

Sensor Probe

- Hot swap connector
- Different sensor probe
- Repalceable



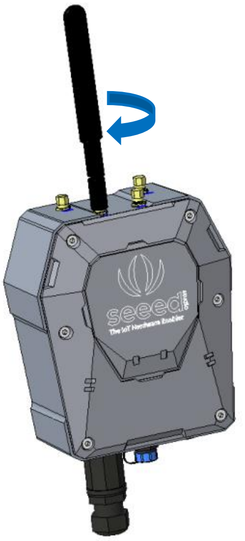


2 Gateway Network Configuration

2.1 The gateway connects to the Internet

2.1.1 Installing Antenna

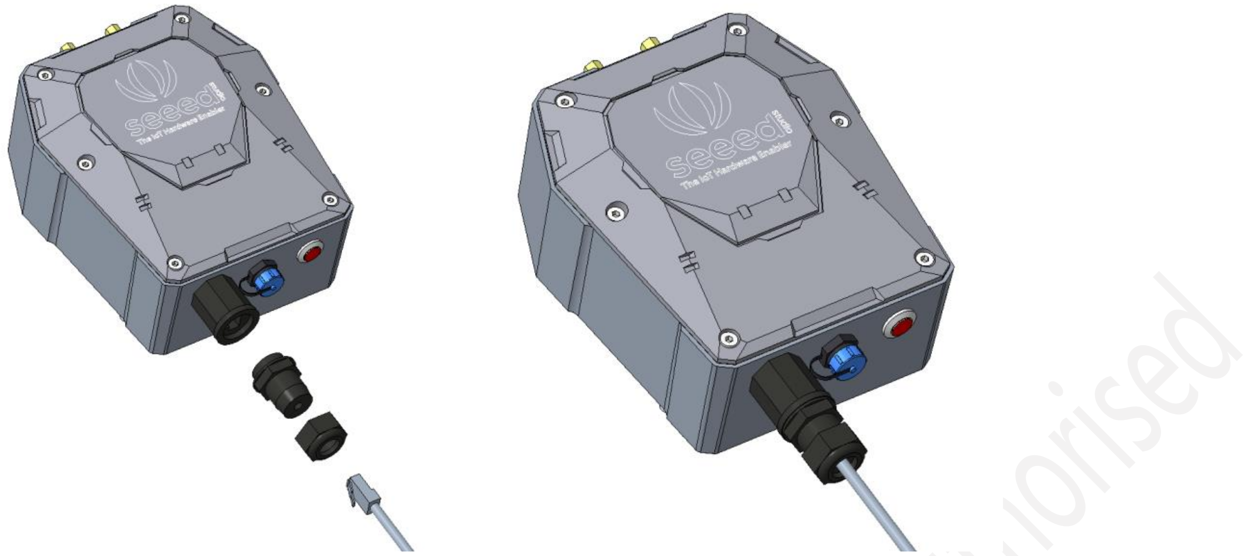
Screw clockwise to install the 4G and LoRa antennas onto the gateway.



2.1.2 Connecting to the Internet

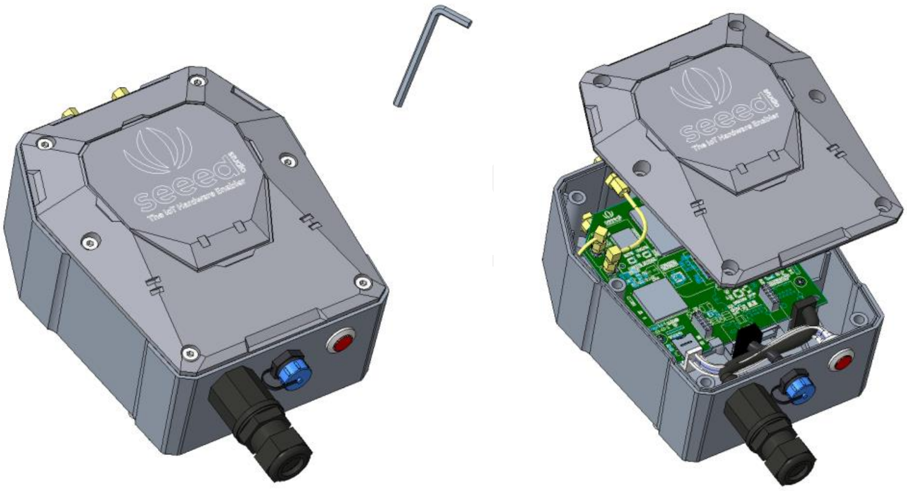
There are two ways to connect to the Internet. Choose the one that works for you.

(1) Connecting to Ethernet Cable
Unscrew to open the protection cap, plug the Ethernet cable through the cap and then into the Ethernet port. Screw to fasten this part.

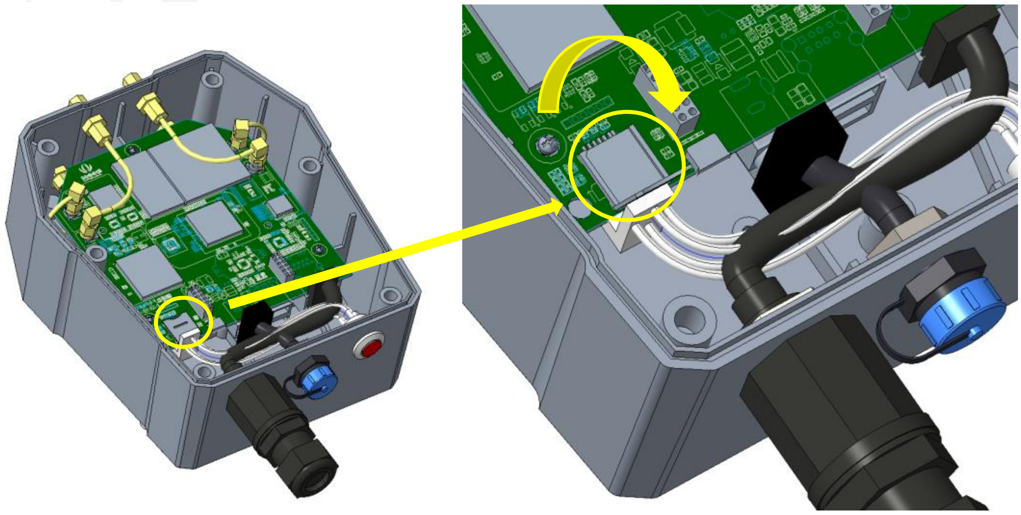


(2) Connecting to 4G

Use the hex key (included in the package) to unscrew the 6 screws and open the lid.

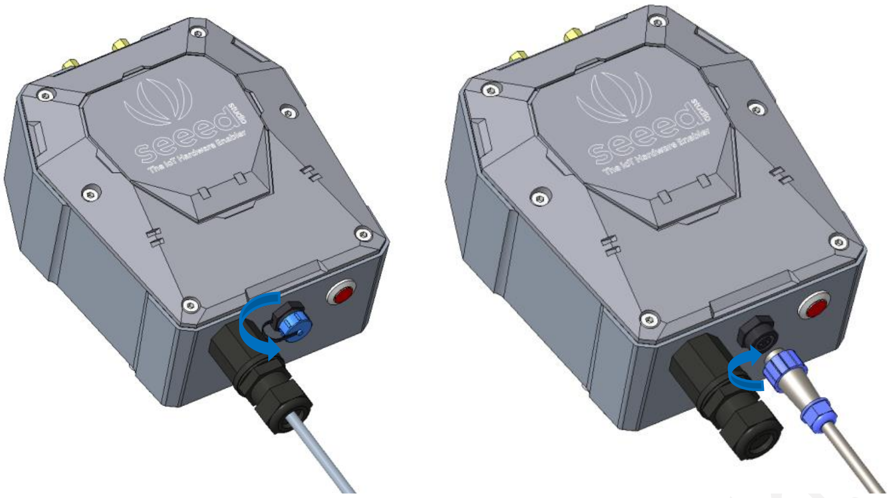


Swipe downward to open the SIM card socket, insert the Micro SIM card and swipe upward to lock the SIM card socket. Make sure it is installed correctly and close the lid with the screws.



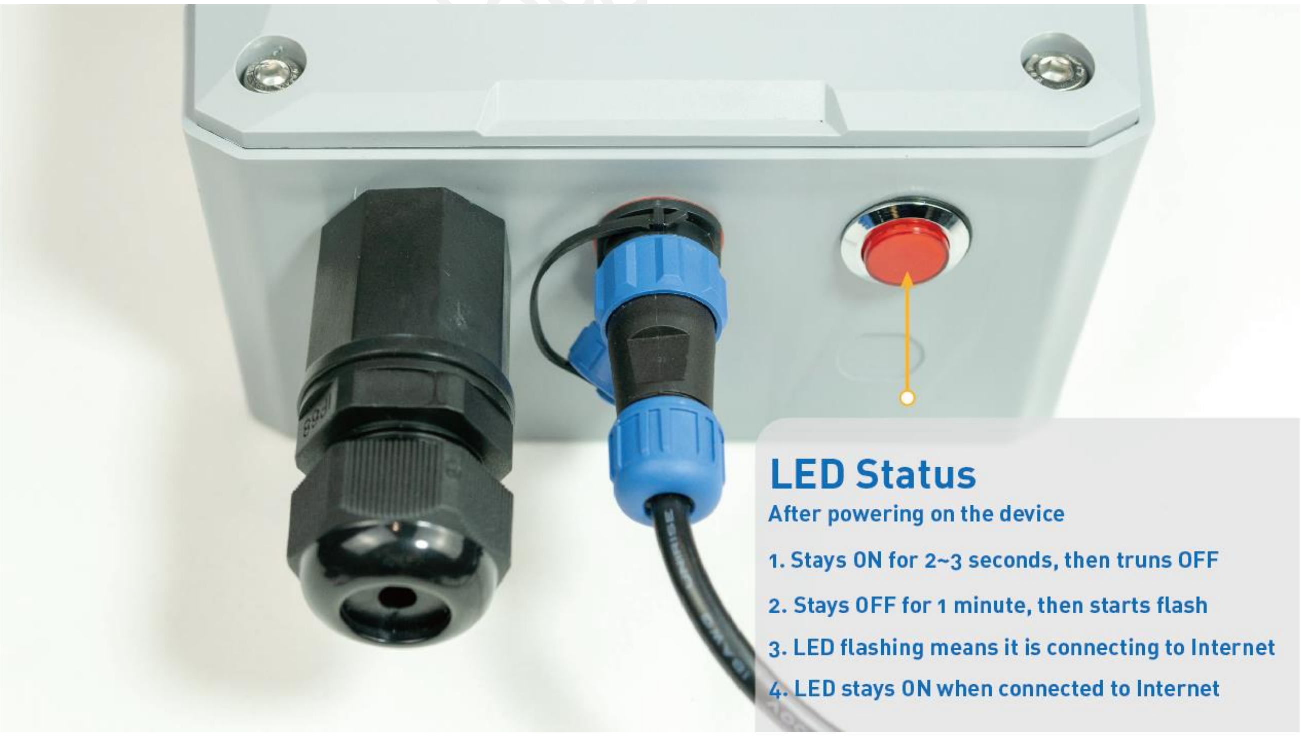
2.1.3 Connecting to Power Cable

Unscrew to take off the power cap, plug in the extension cord and screw to fasten it onto the gateway. The other end of the extension cord is connected to the power adapter.



Notice: Make sure all antennas are correctly installed before powering on the gateway. Please note the device should be POWERED OFF when installing the antenna, or the antenna circuits might be damaged.

2.1.4 The Function of the Red LED



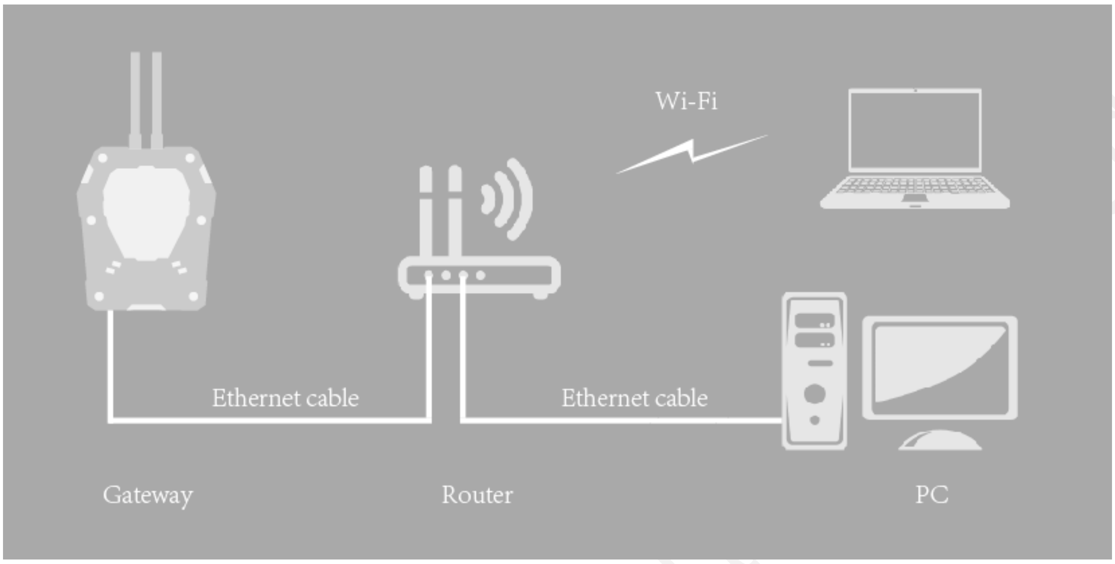
LED Status

After powering on the device

1. Stays ON for 2~3 seconds, then truns OFF
2. Stays OFF for 1 minute, then starts flash
3. LED flashing means it is connecting to Internet
4. LED stays ON when connected to Internet

2.2 Setting the APN

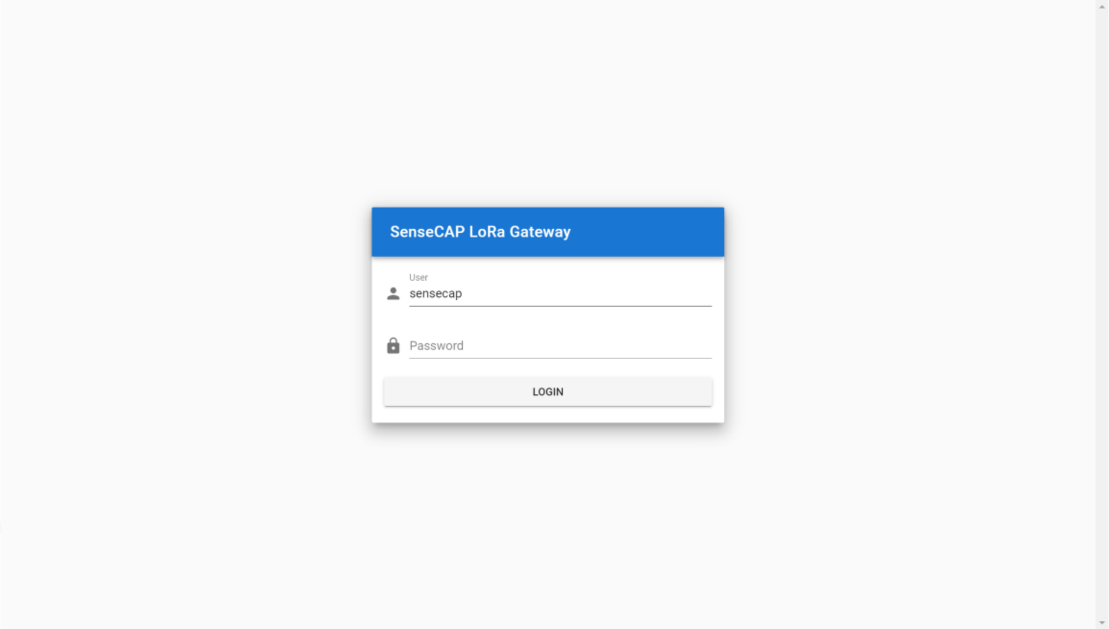
Prepare a router, and the network connection is shown in the figure:



(1) Check the IP of "sensecap" in the background of the router.

(2) Enter IP in the browser: IP:8000

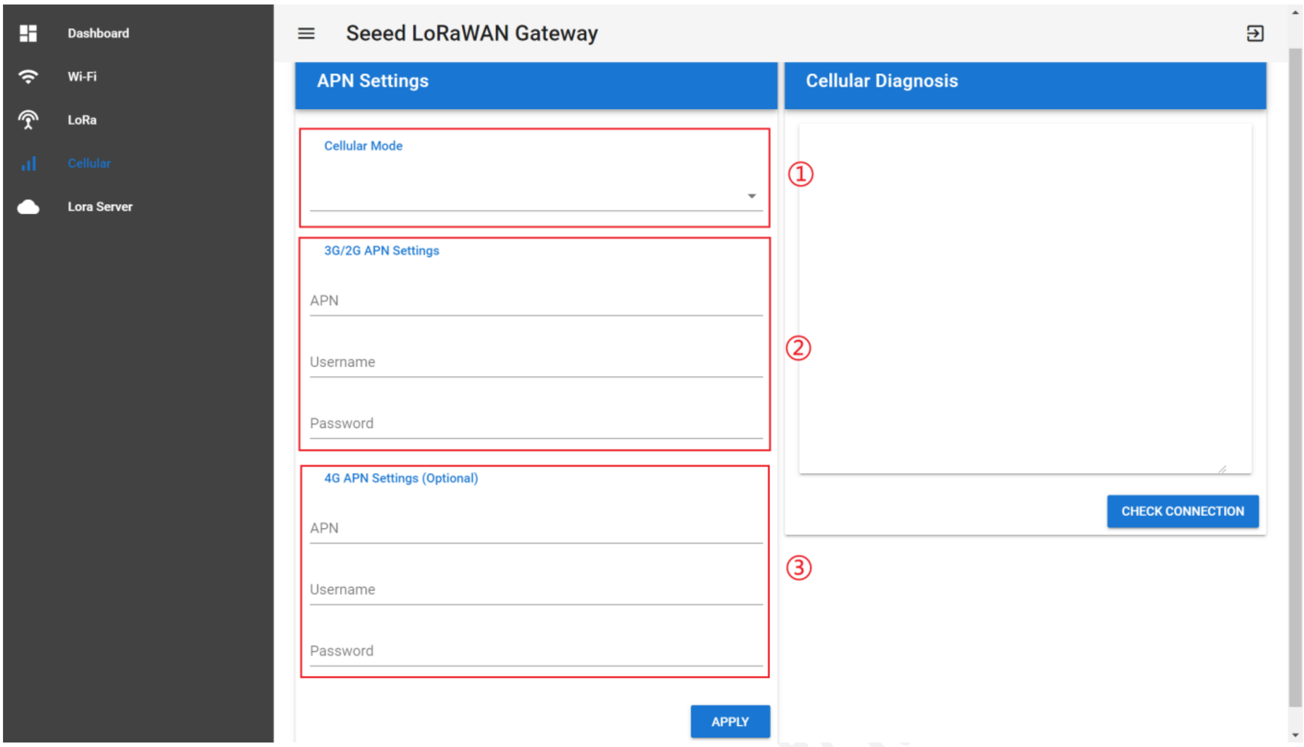
If the IP is 192.168.1.1, enter 192.168.1.1:8000



(3) User: sensecap

Password: sensecap!!!

(4) Click the “Cellular” button.

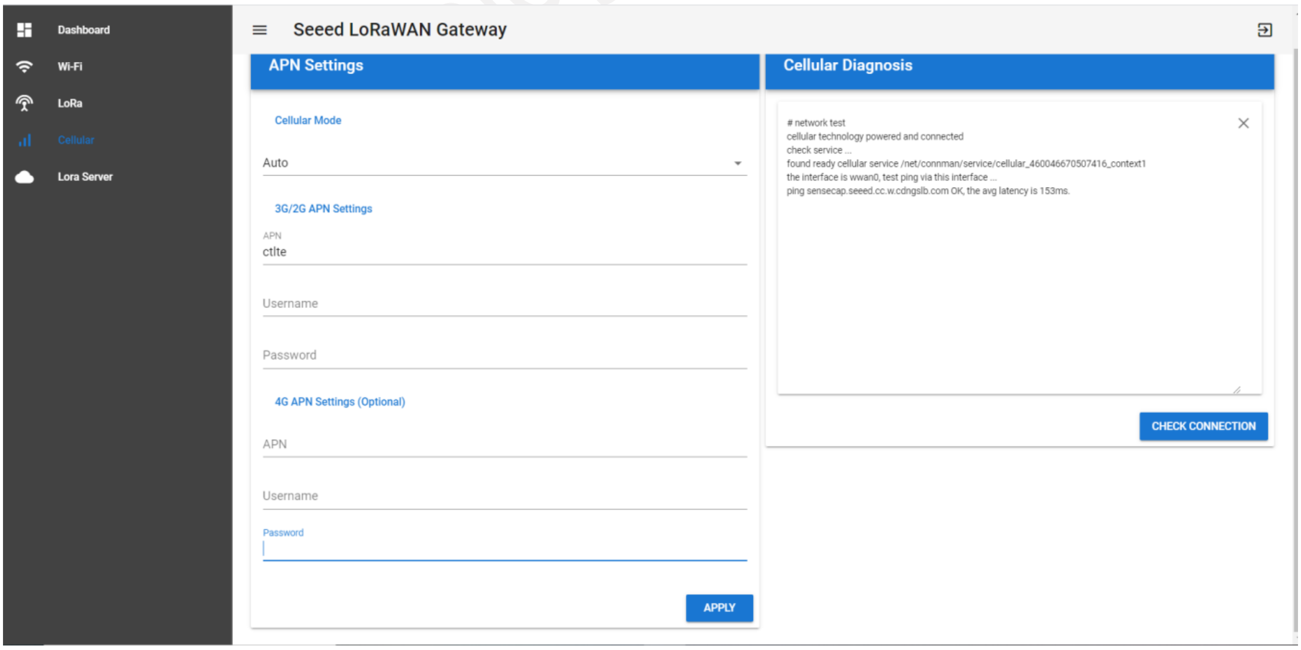


① Cellular Mode: AUTO(default), Gateway automatically selects mode.

② 3G/2G APN Settings: when the mode is 3G/2G, the APN information of SIM card operator needs to be filled in.

③ 4G APN Settings: optional.

(5) Click “APPLY”. Then “CHECK CONNECTION”, if return “cellular technology powered and connected”, it means ok.



Cellular Mode: Auto

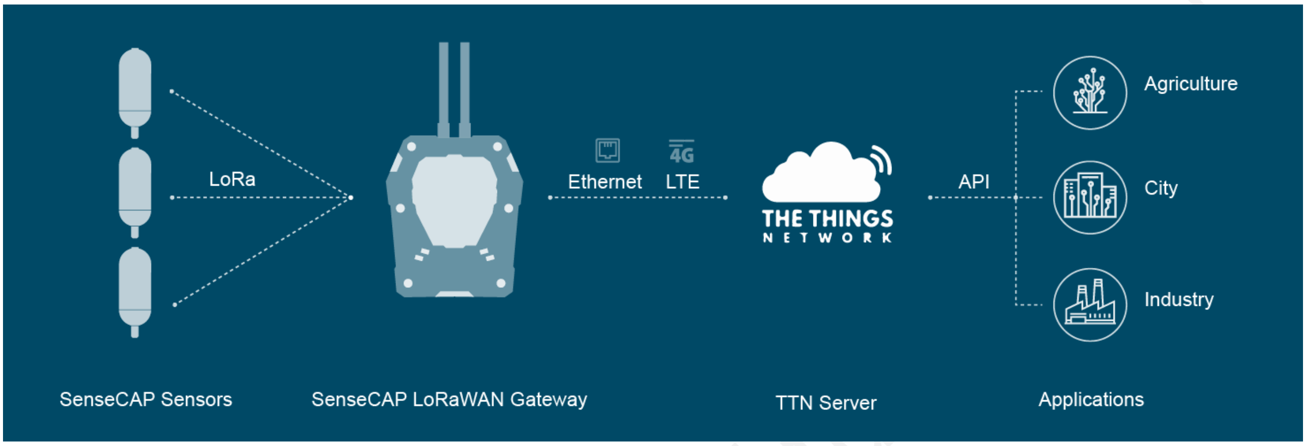
APN: ctlte

```
# network test
cellular technology powered and connected
check service ...
found ready cellular service /net/connman/service/cellular_460046670507416_context1
the interface is wwan0, test ping via this interface ...
ping sensecap.seeed.cc.w.cdngslb.com OK, the avg latency is 153ms.
```

3 Add Gateway to User’s TTN Server

The SenseCAP LoRaWAN Gateway supports connecting to the user's own The Things Network account and server.

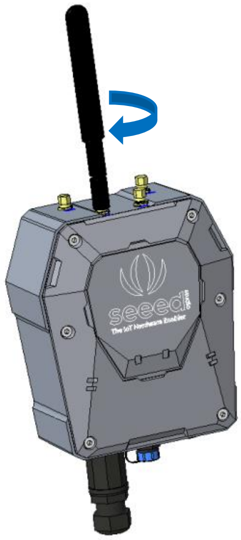
Learn more about TTN: https://www.thethingsnetwork.org/docs/



3.1 Gateway Network Configuration

3.1.1 Installing Antenna

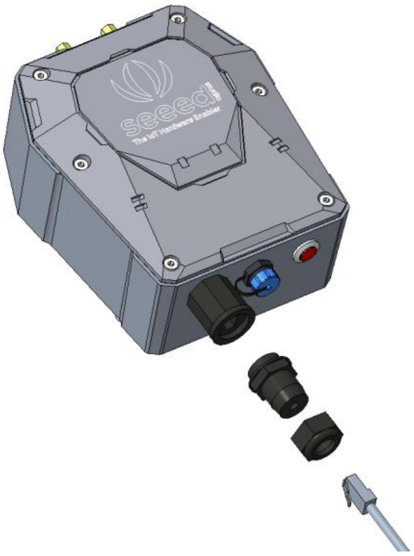
Screw clockwise to install the 4G and LoRa antennas onto the gateway.



3.1.2 Connecting to the Internet

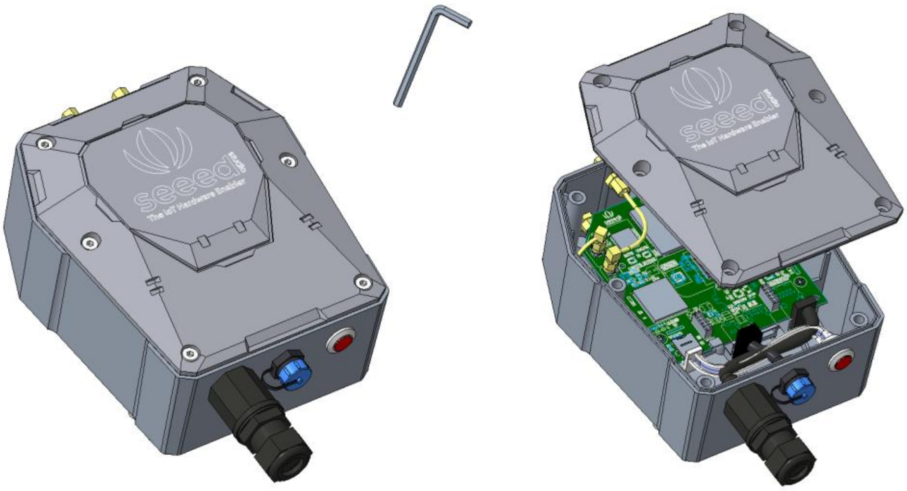
There are two ways to connect to the Internet. Choose the one that works for you.

(3) Connecting to Ethernet Cable
Unscrew to open the protection cap, plug the Ethernet cable through the cap and then into the Ethernet port. Screw to fasten this part.

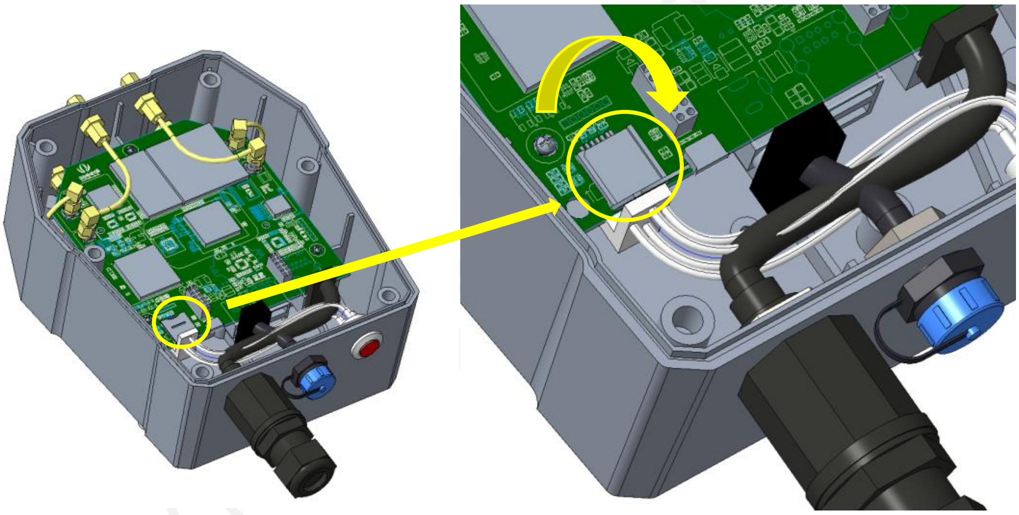


(4) Connecting to 4G

Use the hex key (included in the package) to unscrew the 6 screws and open the lid.

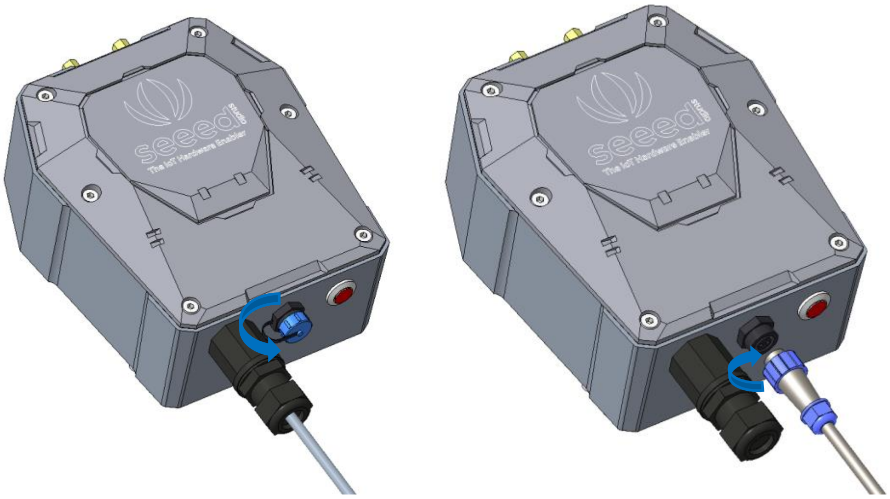


Swipe downward to open the SIM card socket, insert the Micro SIM card and swipe upward to lock the SIM card socket. Make sure it is installed correctly and close the lid with the screws.



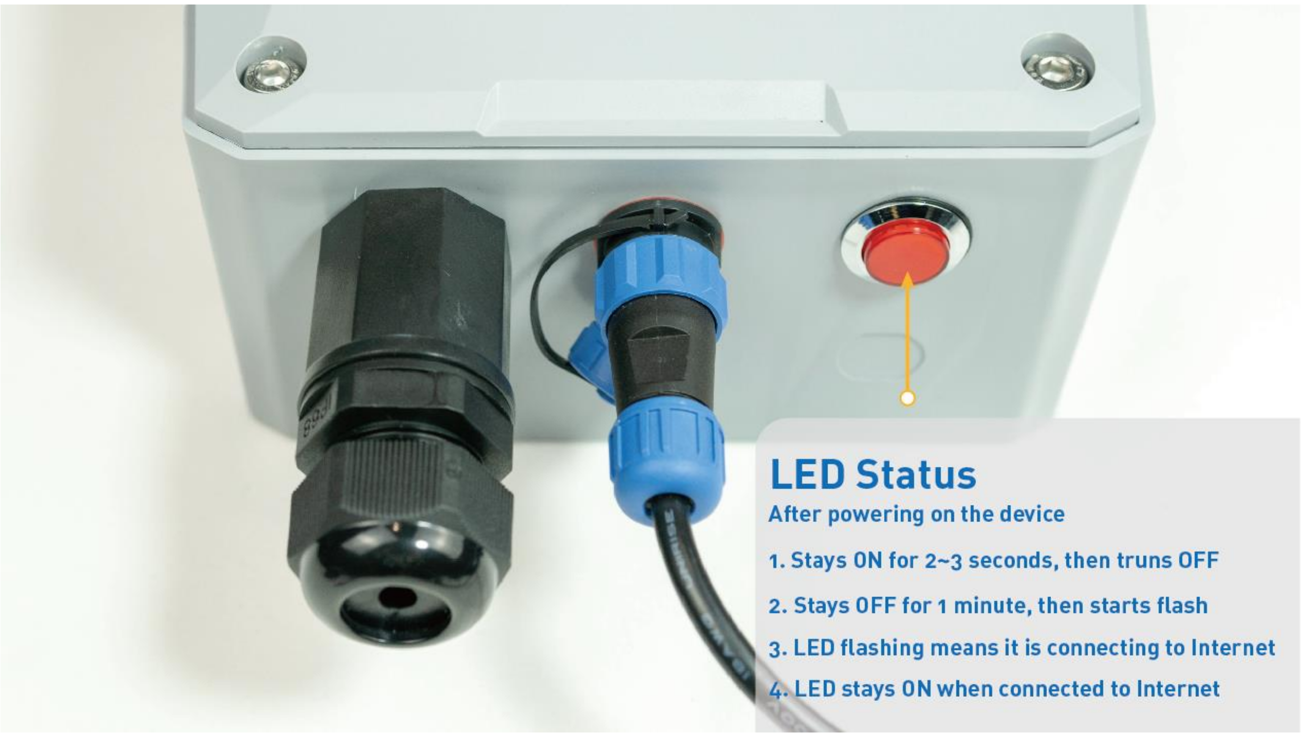
3.1.3 Connecting to Power Cable

Unscrew to take off the power cap, plug in the extension cord and screw to fasten it onto the gateway. The other end of the extension cord is connected to the power adapter.



Notice: Make sure all antennas are correctly installed before powering on the gateway. Please note the device should be POWERED OFF when installing the antenna, or the antenna circuits might be damaged.

3.1.4 The Function of the Red LED



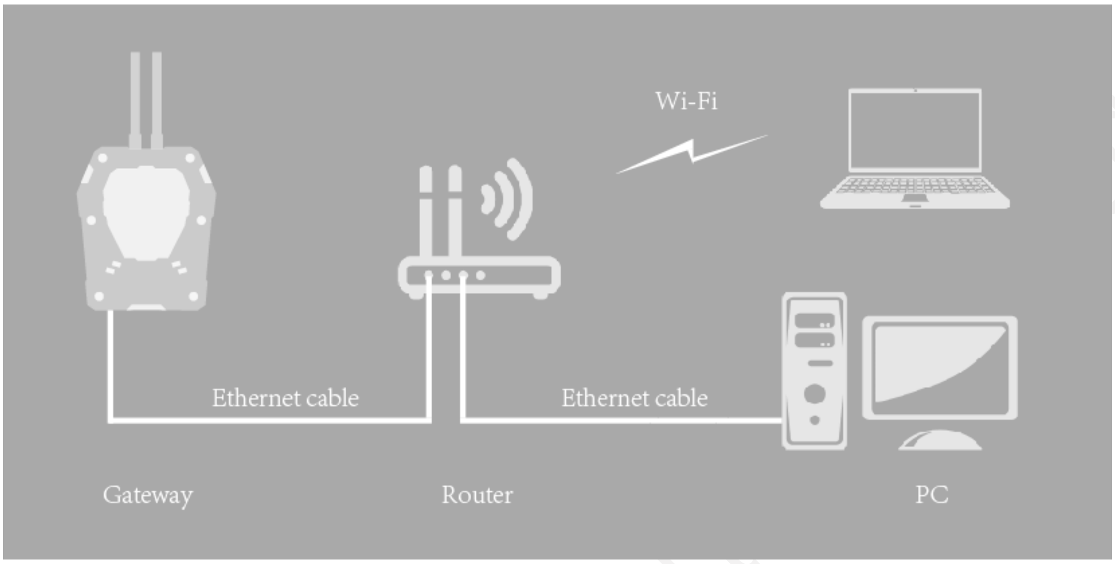
LED Status

After powering on the device

1. Stays ON for 2~3 seconds, then truns OFF
2. Stays OFF for 1 minute, then starts flash
3. LED flashing means it is connecting to Internet
4. LED stays ON when connected to Internet

3.2 Setting the Gateway Service Address

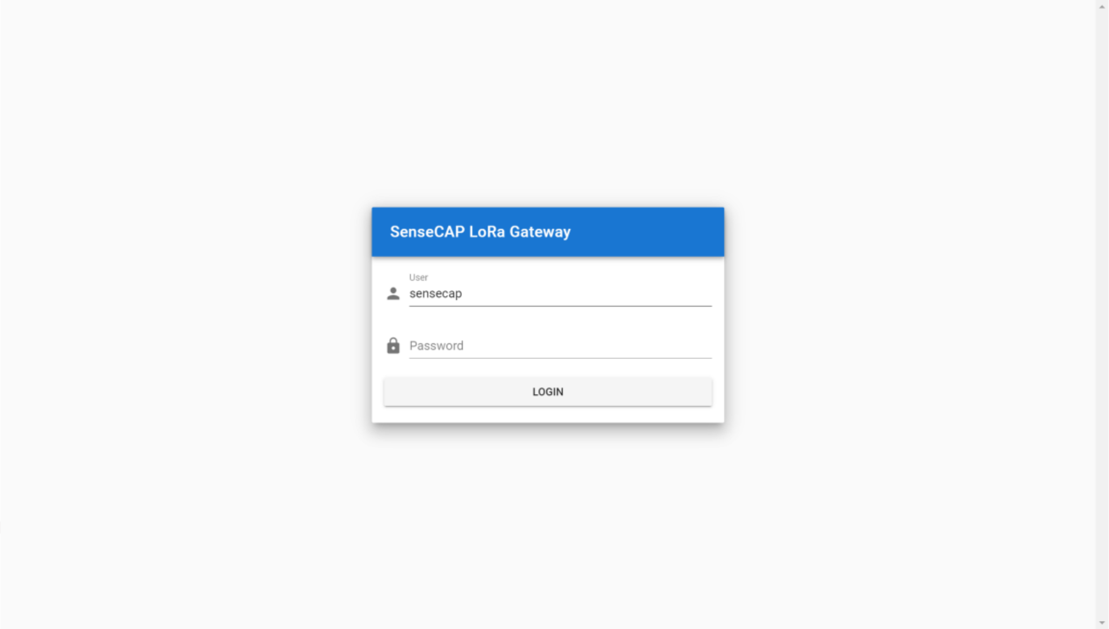
Prepare a router, and the network connection is shown in the figure:



(6) Check the IP of "sensecap" in the background of the router.

(7) Enter IP in the browser: IP:8000

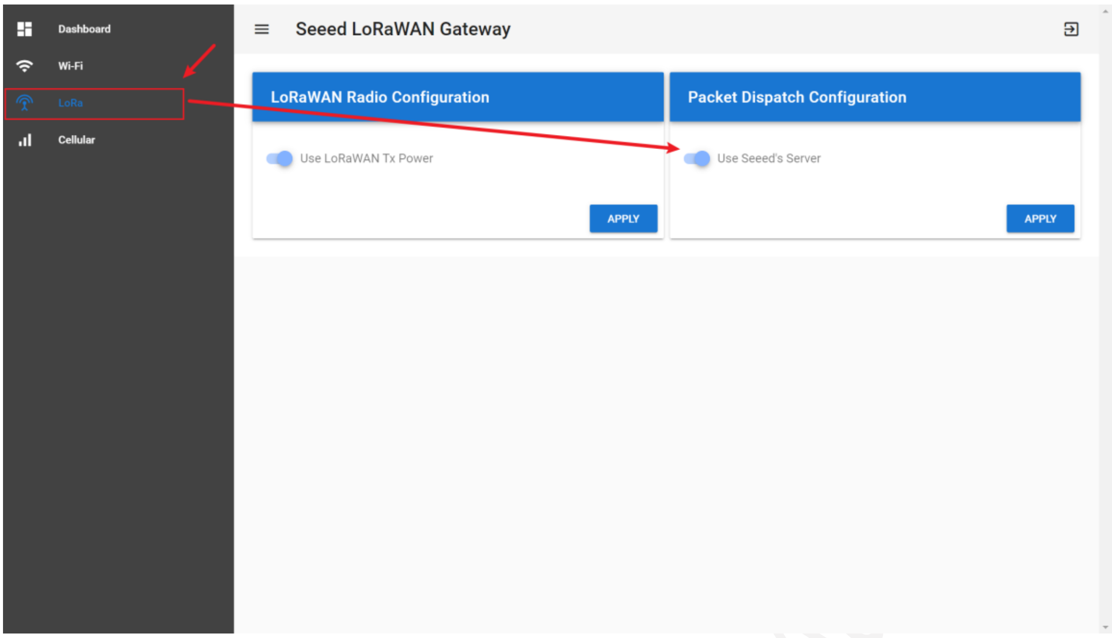
If the IP is 192.168.1.1, enter 192.168.1.1:8000



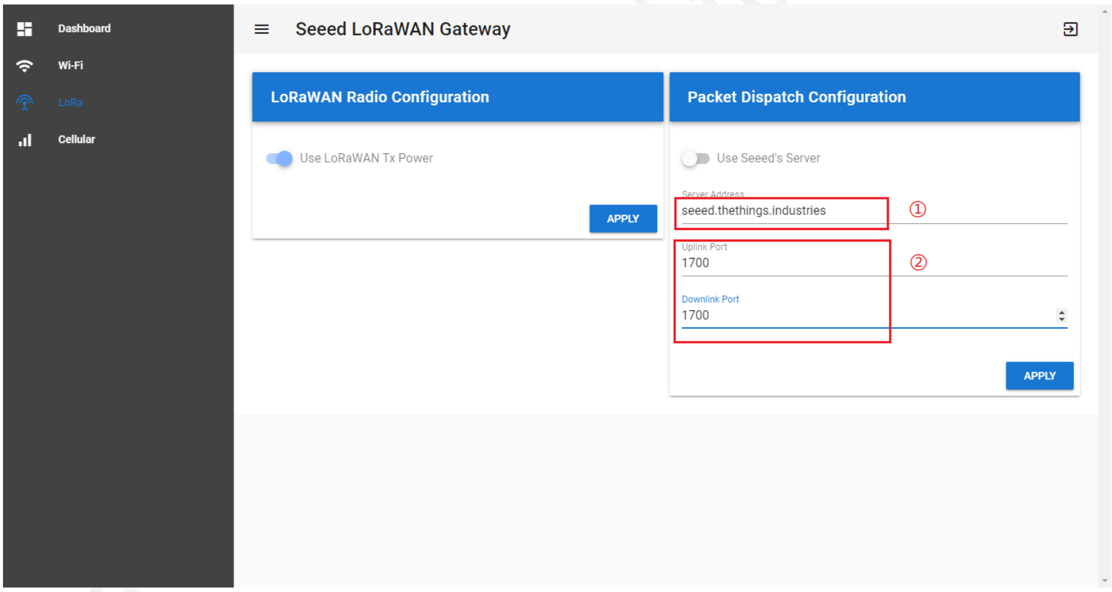
(8) User: sensecap

Password: sensecap!!!

(9) LoRa→Use Seeed's Server→Off Button



(10)



Server Address: seeed.thethings.industries ①

Uplink Port: 1700 ②

Downlink Port: 1700

① Server Address: Please input your Server Address.

Refer to the table or website: https://www.thethingsnetwork.org/docs/gateways/packet-forwarder/semtech-udp.html#router-addresses

Router address	Region
router.eu.thethings.network	EU 433 and EU 863-870
router.us.thethings.network	US 902-928
router.cn.thethings.network	China 470-510 and 779-787
router.as.thethings.network	Southeast Asia 923 MHz

router.as1.thethings.network	Southeast Asia 920-923 MHz
router.as2.thethings.network	Southeast Asia 923-925 MHz
router.kr.thethings.network	Korea 920-923 MHz
router.jp.thethings.network	Japan 923-925 MHz (with EIRP cap according to Japanese regulations)
thethings.meshed.com.au	Australia 915-928 MHz
as923.thethings.meshed.com.au	Australia (Southeast Asia 923MHz frequency plan)
ttn.opennetworkinfrastructure.org	Switzerland (EU 433 and EU 863-870)

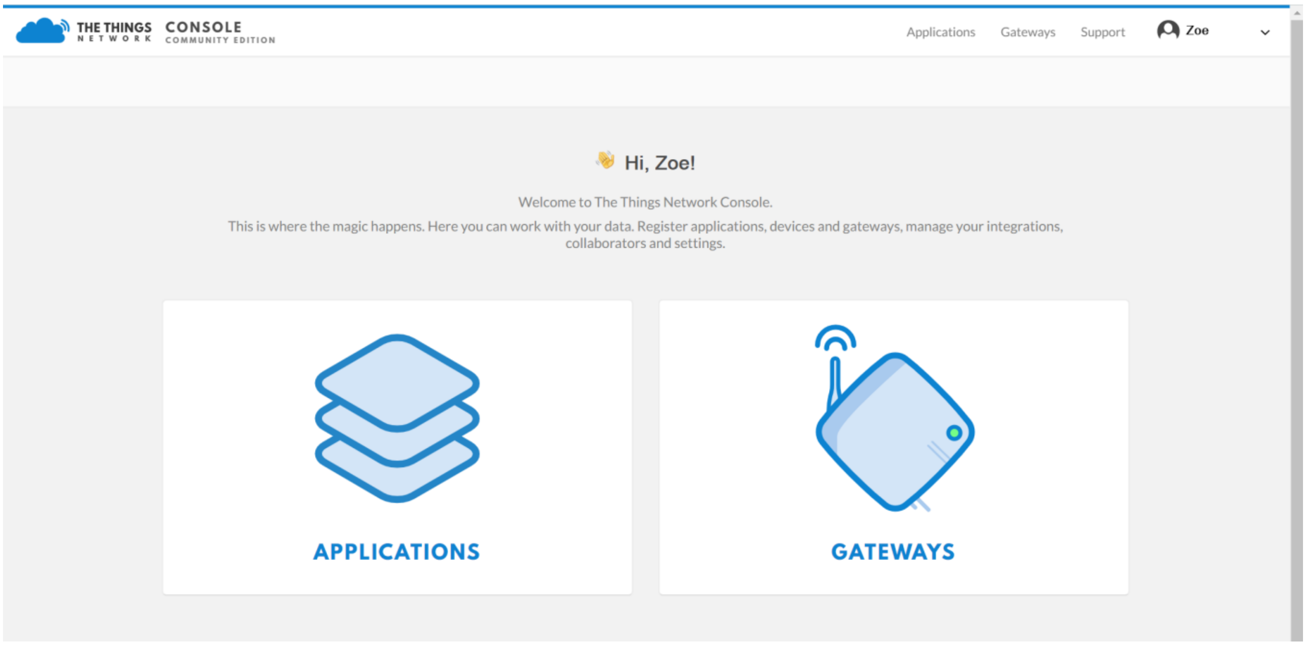
② Uplink / Downlink Port (default): 1700

(11) APPLY.

3.3 Gateway Registration on TTN

TTN website: https://www.thethingsnetwork.org

(1) Follow the instruction to create your account, and access "Console".

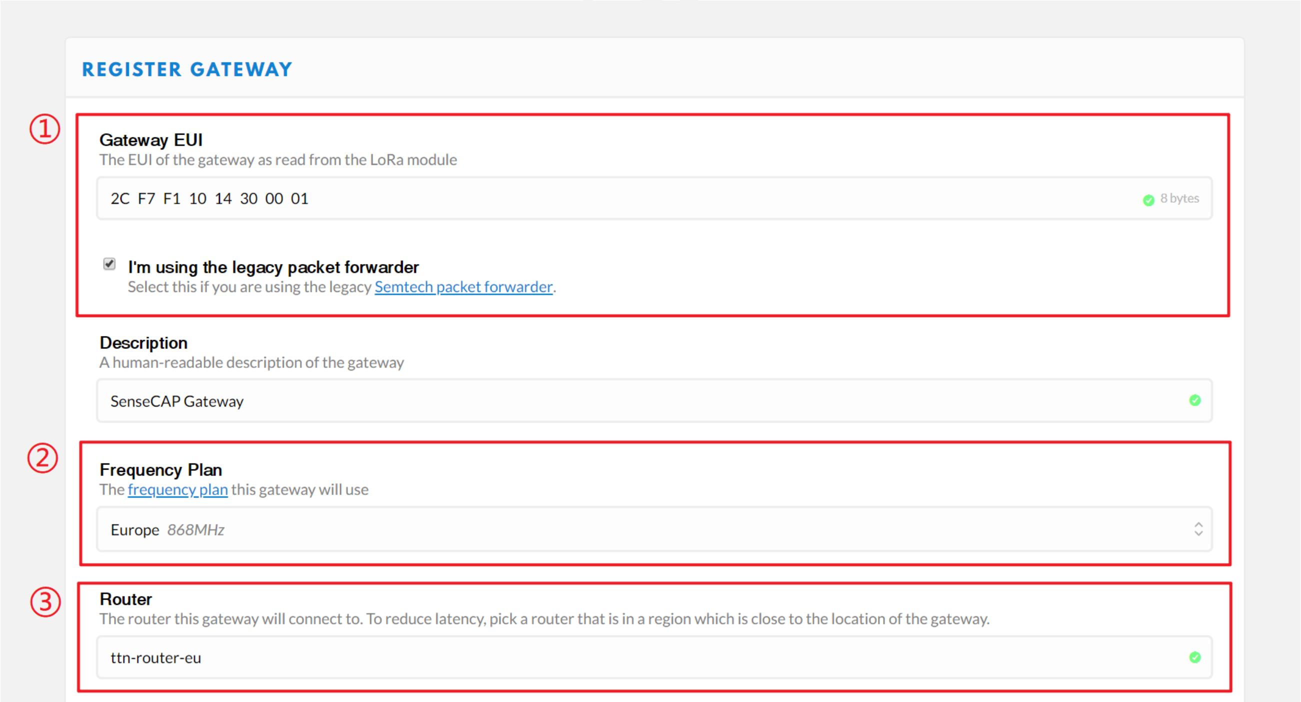


Hi, Zoe!

Welcome to The Things Network Console.

This is where the magic happens. Here you can work with your data. Register applications, devices and gateways, manage your integrations, collaborators and settings.

(2) Register Gateway



REGISTER GATEWAY

① Gateway EUI
The EUI of the gateway as read from the LoRa module
2C F7 F1 10 14 30 00 01 — 8 bytes

☑ I'm using the legacy packet forwarder
Select this if you are using the legacy Semtech packet forwarder.

Description
A human-readable description of the gateway
SenseCAP Gateway

② Frequency Plan
The frequency plan this gateway will use
Europe 868MHz

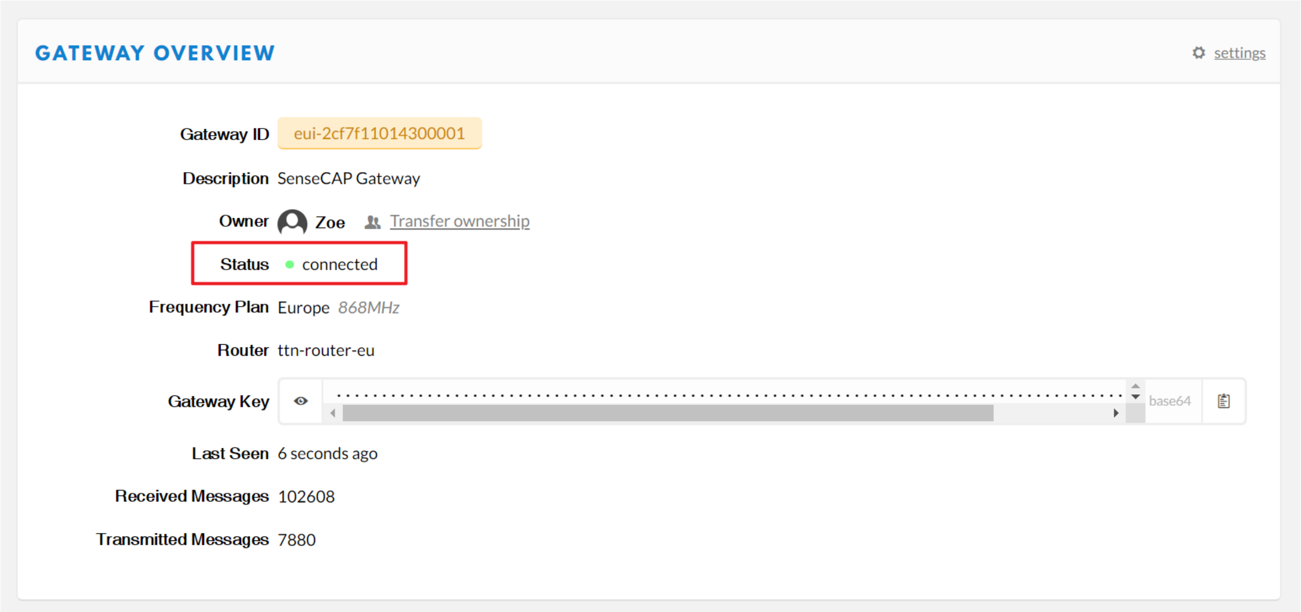
③ Router
The router this gateway will connect to. To reduce latency, pick a router that is in a region which is close to the location of the gateway.
ttn-router-eu

① Gateway EUI: View the labels on the gateway.
Select 'I'm using the legacy packet forwarder'.

② Frequency Plan: View the labels on the gateway.

③ Router: Select the router that is right for you.

④ Register.
Gateway Status displays connected, indicating successful registration.



GATEWAY OVERVIEW

Gateway ID eui-2cf7f11014300001

Description SenseCAP Gateway

Owner Zoe Transfer ownership

Status ● connected

Frequency Plan Europe *868MHz*

Router ttn-router-eu

Gateway Key base64

Last Seen 6 seconds ago

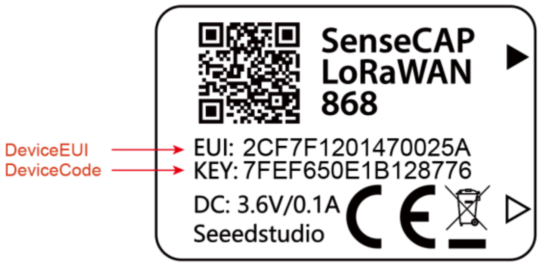
Received Messages 102608

Transmitted Messages 7880

4 Add Sensor Node to User’s TTN Server

4.1 Get Node’s EUI and Key

(1) DeviceEUI and DeviceCode is on the SenseCAP product label.



(2) SenseCAP sensor device’s AppEUI and AppKey have been flash into the device by Seeed. Use HTTP API to retrieve App EUI and App Key. You can use browser to issue an HTTP GET request.

Curl:

```
https://sensecap.seeed.cc/makerapi/device/view_device_info?nodeEui=2CF7F12014700297&deviceCode=34BF25920A4EFBF4
```

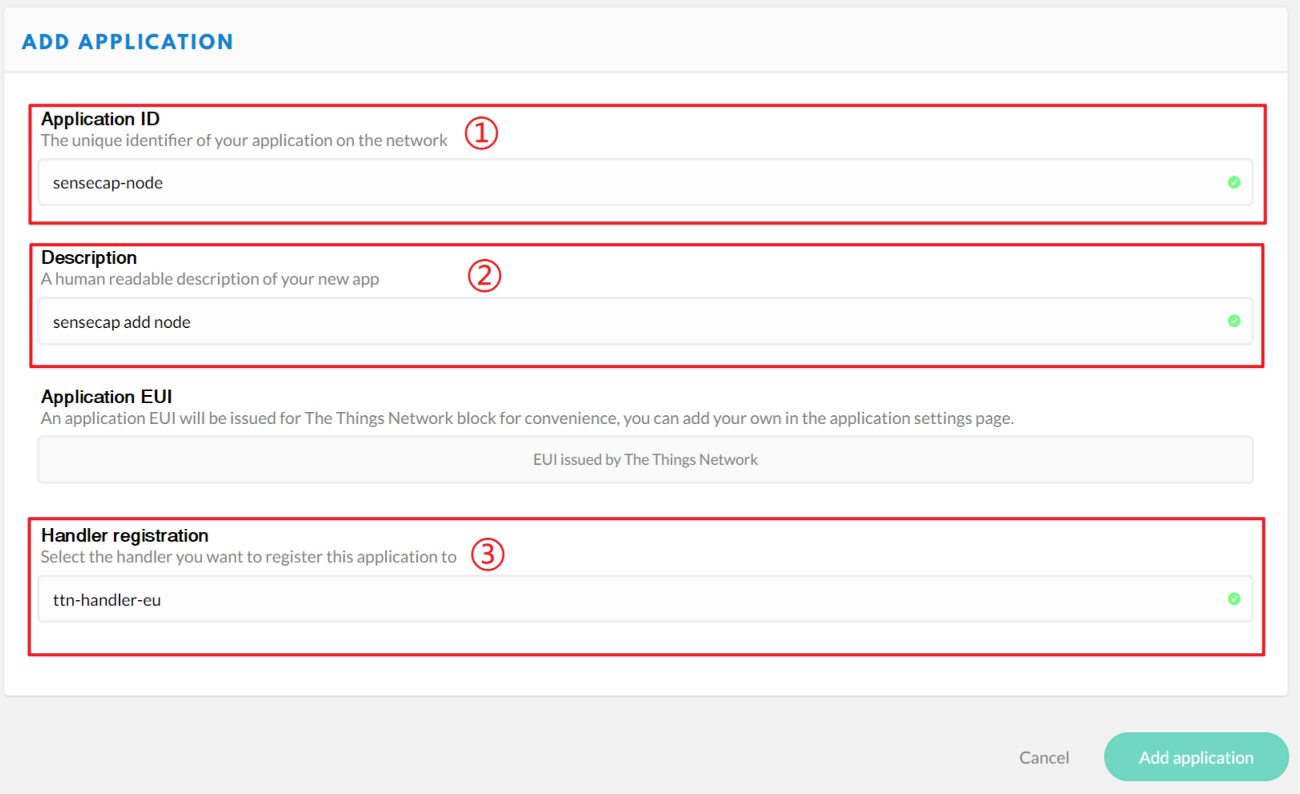
In the API, replace the DeviceEUI and deviceCode with your own DeviceEUI and DeviceCode respectively. And you will get the following response.

```
{
   "code": "0",
   "data": {
      "nodeEui": "2CF7F12014700297",
      "deviceCode": "34BF25920A4EFBF4",
      "lorawanInformation": {
         "dev_eui": "2CF7F12014700297",
         "app_eui": "8000000000000006",
         "app_key": "6FD0EF47CBC6E00F1921A08C2E94E8E5"
      }
   },
   "time": 0.019
}
```

4.2 Add Application and AppEUI

(1) TTN console → Application → Add application

(2)



ADD APPLICATION

Application ID
The unique identifier of your application on the network ①
sensecap-node

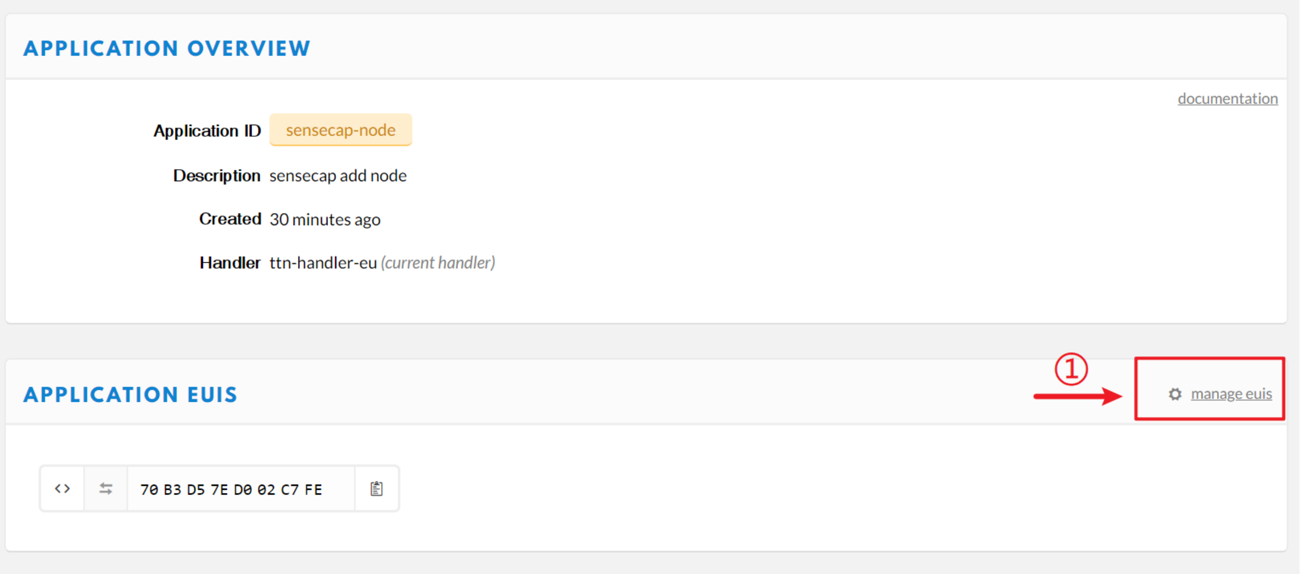
Description
A human readable description of your new app ②
sensecap add node

Application EUI
An application EUI will be issued for The Things Network block for convenience, you can add your own in the application settings page.
EUI issued by The Things Network

Handler registration
Select the handler you want to register this application to ③
ttn-handler-eu

① Application ID: Enter a unique name.
② Description: Enter a description.
③ Handler registration: Select the same handler as the gateway router.
④ Add application.

(3)

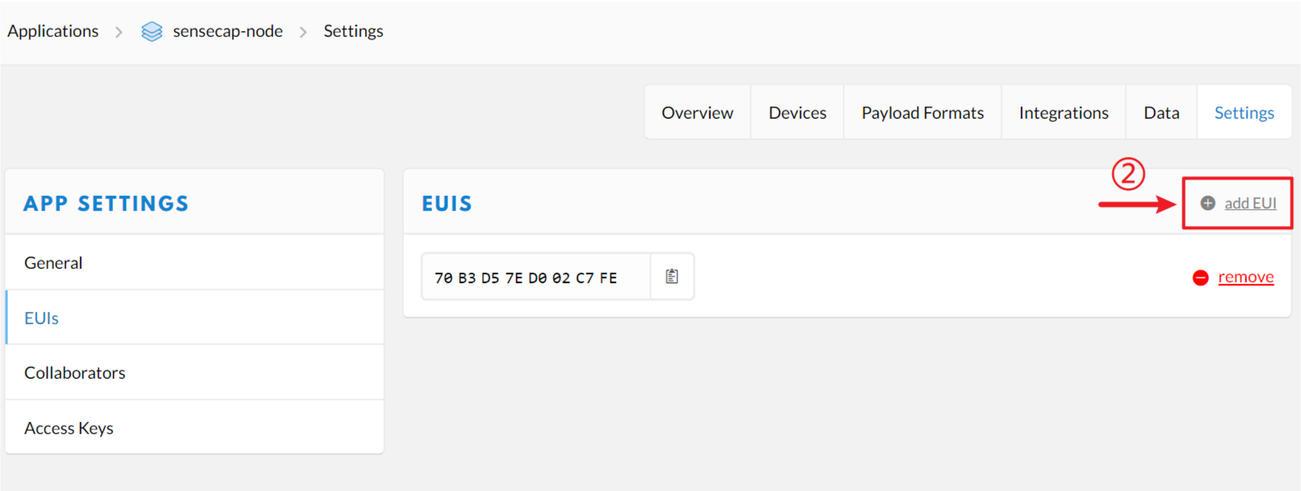


APPLICATION OVERVIEW

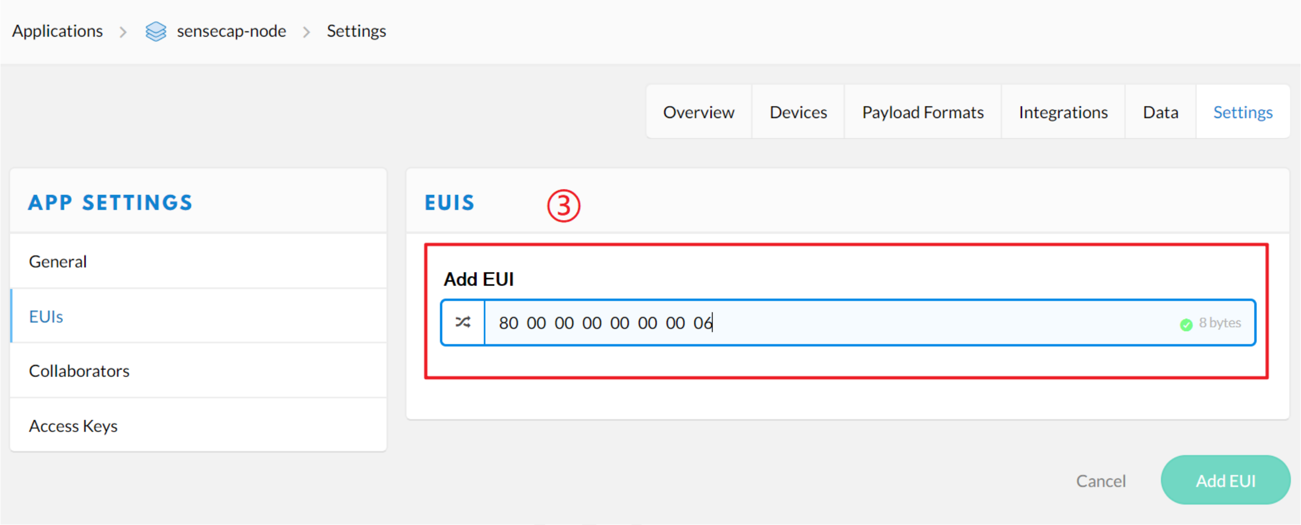
Application ID sensecap-node
Description sensecap add node
Created 30 minutes ago
Handler ttn-handler-eu *(current handler)*

APPLICATION EUIS

70 B3 D5 7E D0 02 C7 FE



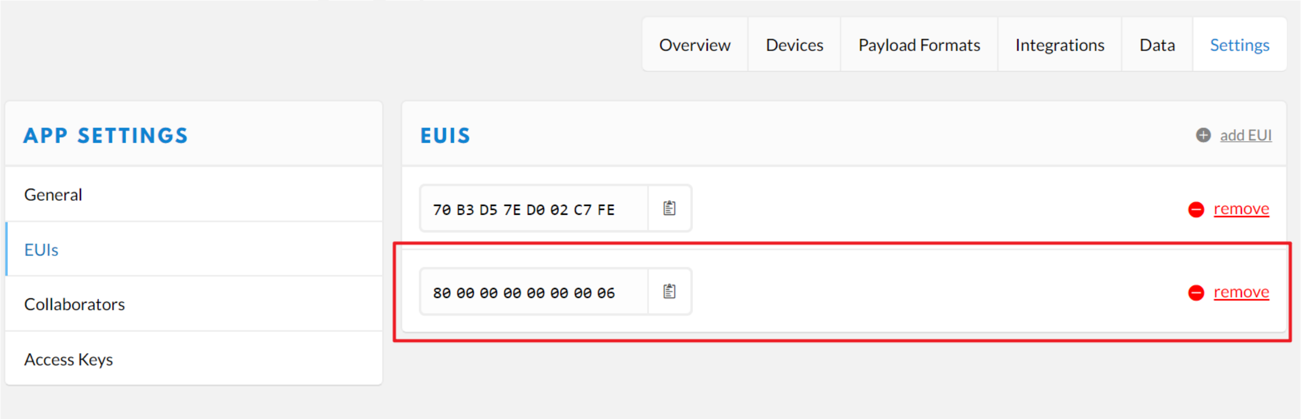
70 B3 D5 7E D0 02 C7 FE



Add EUI

80 00 00 00 00 00 00 06

① Application → Application EUIS → Manage EUIs.
② →Add EUI.
③ Enter the node's AppEui that you got in the 3.1 step.
④ →Add EUI.

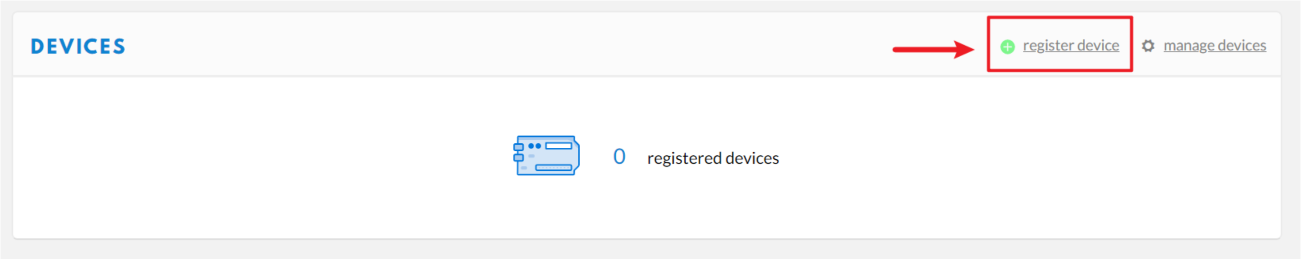


70 B3 D5 7E D0 02 C7 FE

80 00 00 00 00 00 00 06

4.3 Sensor Node Registration on TTN

(1) Application → Devices → register device



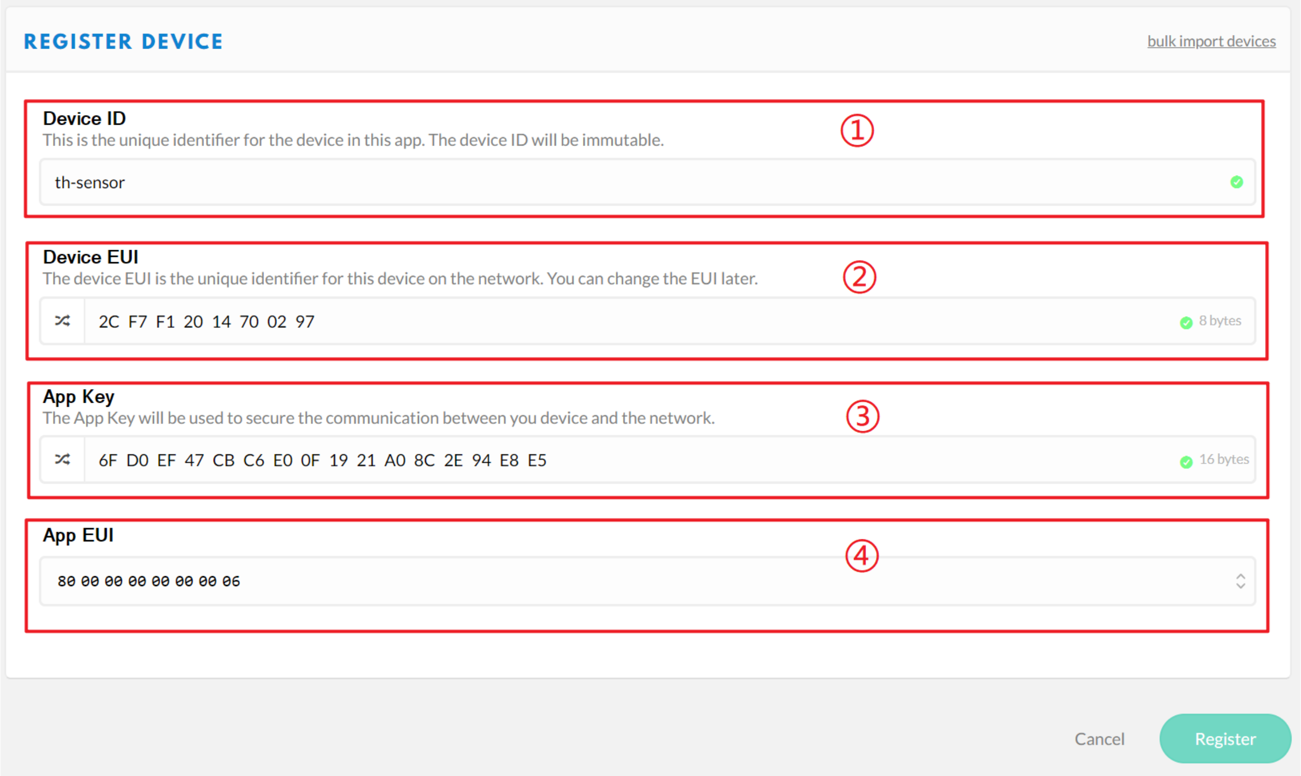
DEVICES

register device

manage devices

0 registered devices

(2)



REGISTER DEVICE

bulk import devices

Device ID ①

This is the unique identifier for the device in this app. The device ID will be immutable.

th-sensor

Device EUI ②

The device EUI is the unique identifier for this device on the network. You can change the EUI later.

2C F7 F1 20 14 70 02 97 — 8 bytes

App Key ③

The App Key will be used to secure the communication between you device and the network.

6F D0 EF 47 CB C6 E0 0F 19 21 A0 8C 2E 94 E8 E5 — 16 bytes

App EUI ④

80 00 00 00 00 00 00 06

Cancel Register

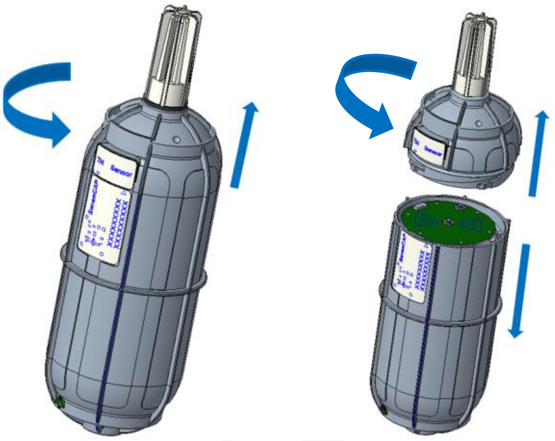
① Device ID: Enter a unique name.
② Device EUI: Enter the node’s Device EUI that you got in the 3.1 step.
③ App Key: Enter the node’s App Key that you got in the previous step.
④ App EUI: Select the node’s App EUI.
⑤ Register.

4.4 Connect the Node to TTN

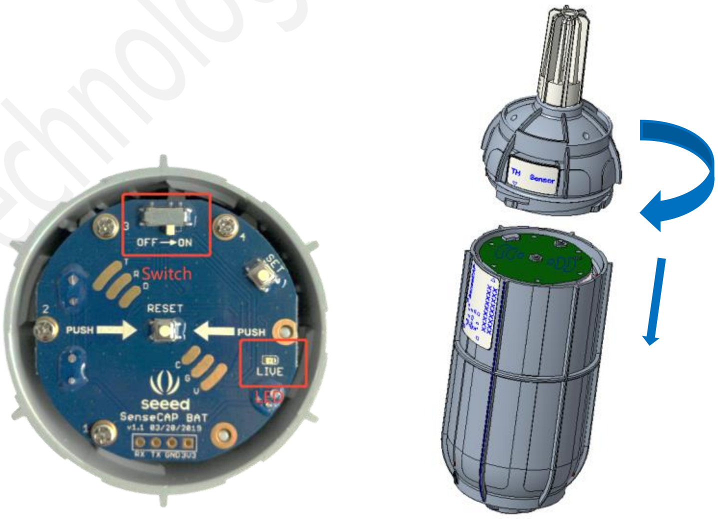
4.4.1 Power on

The power switch is hidden inside the device. Open the device and turn on the power before installing the sensors. Here is the step-by-step instruction:

1) Loosen the Sensor Probe by turning the cap counterclockwise. Use the white cap opener to make this process easier. The image below uses TH Sensor as an example and applies to all other SenseCAP sensors.



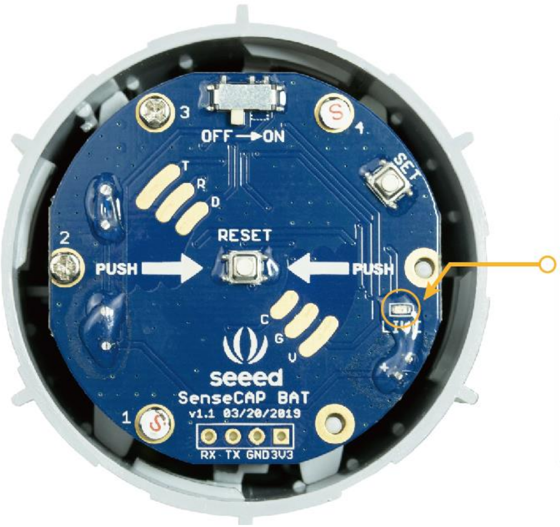
2) After opening the device, turn the switch to "ON", and the LED on the lower right corner will flash, indicating that the power is on. Wait for about 10 seconds, then the LED will flash quickly for 2 seconds, indicating that the device is connected to the network.



3) After the device is connected to the network, connect the Sensor Probe back with the Sensor Node Controller by turning it clockwise. Please note that the labels on both parts should be aligned as shown in the image below, otherwise the two parts will not be attached to function properly and data will not be uploaded.

4.4.2 Sensor Node Working Status

You can refer to the LED indicator for the Sensor Node for its working status. Please see the status explanations in the image below:



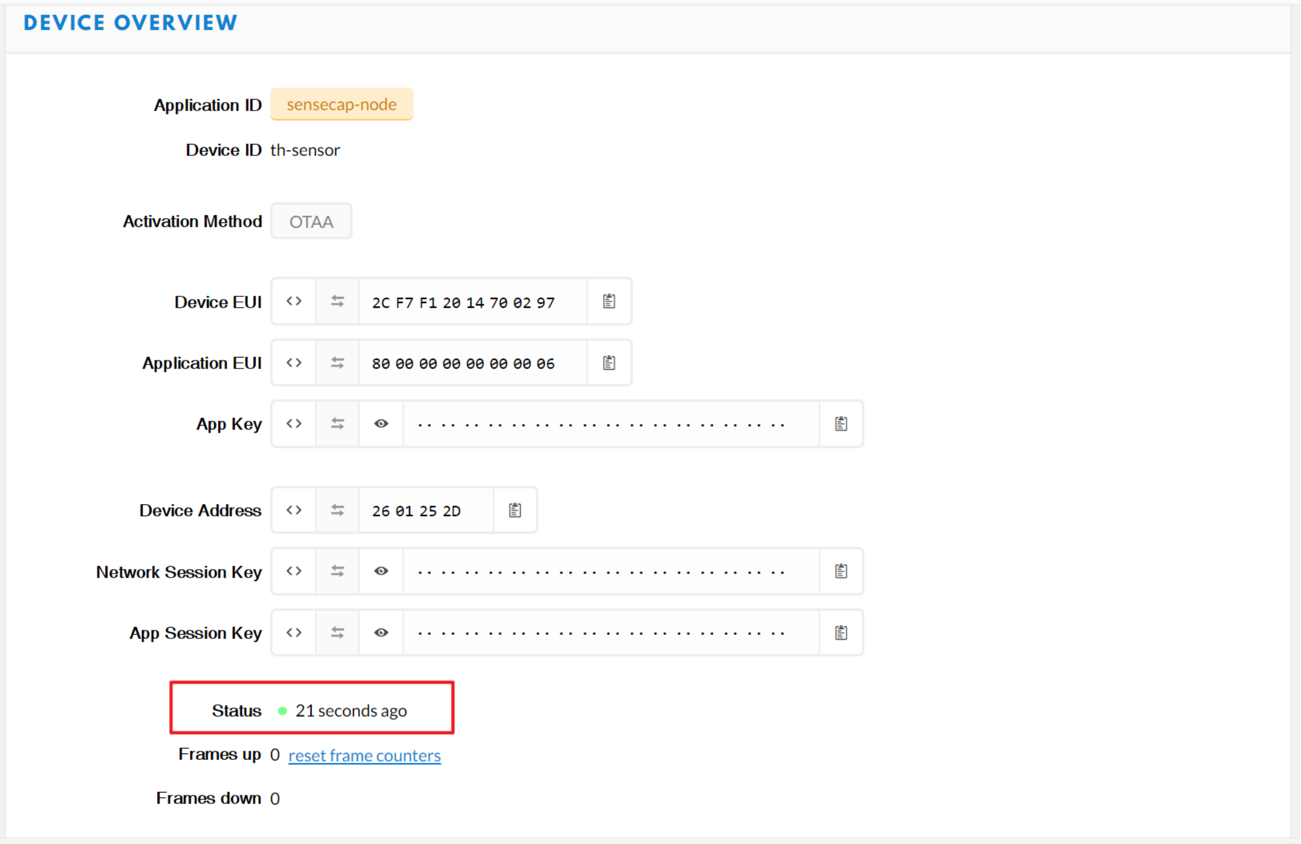
LED Status

After powering on the device

1. LED flashes once after powering on, then turn OFF
2. After 10 seconds, LED flashes quickly for 2 seconds, indicating it has joined the network
3. After joining the network, the LED stays off to save energy
4. Push the reset button to re-join the network if the LED does not start flashing 15 seconds after powering on

4.4.3 Checking Sensor Node Connection to the TTN

(1) On the Device Overview page, Status turns green.



DEVICE OVERVIEW

Application ID sensecap-node

Device ID th-sensor

Activation Method OTAA

Device EUI 2C F7 F1 20 14 70 02 97

Application EUI 80 00 00 00 00 00 00 06

App Key ·

Device Address 26 01 25 2D

Network Session Key ·

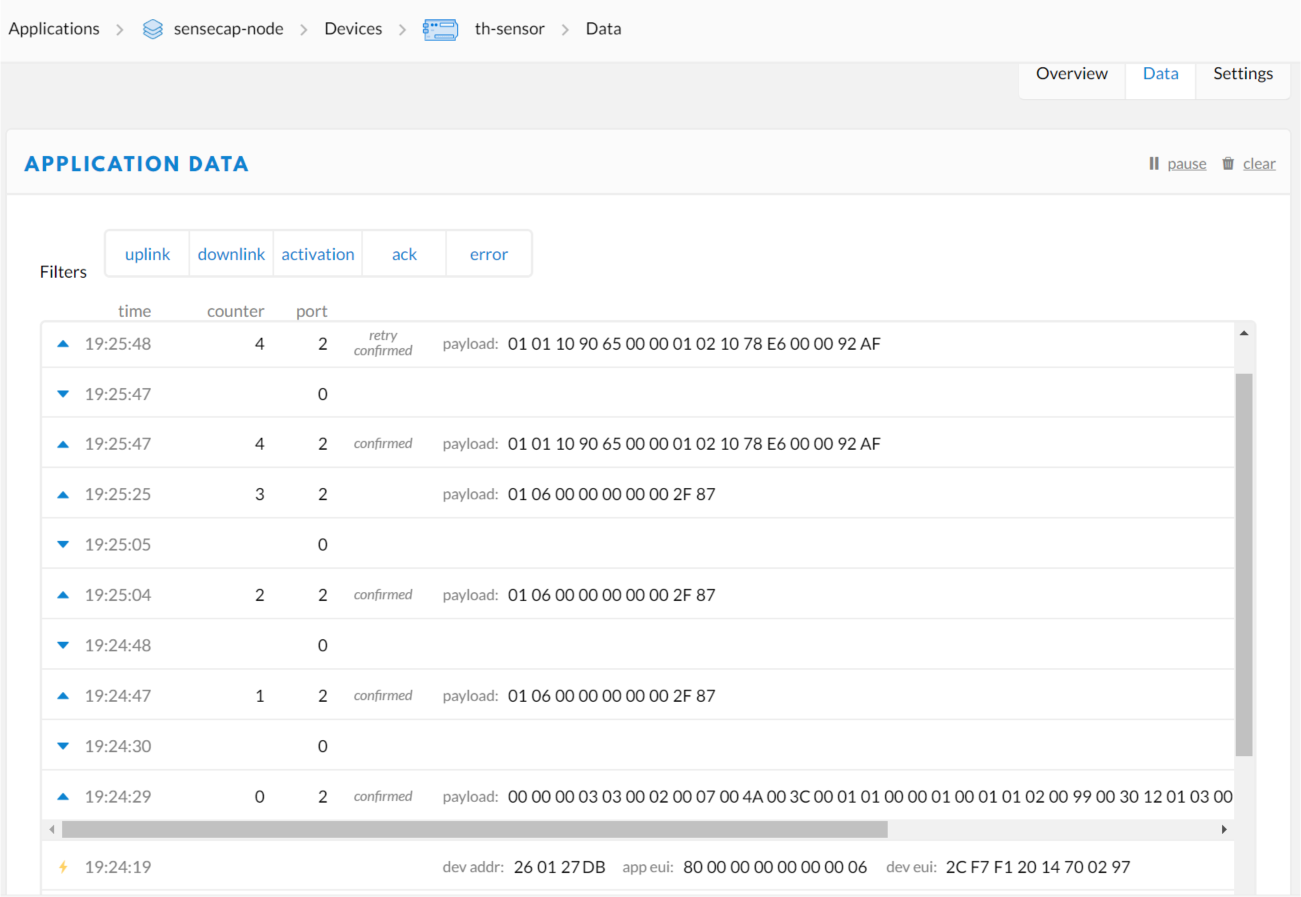
App Session Key ·

Status ● 21 seconds ago

Frames up 0 reset frame counters

Frames down 0

(2) On the Data page, data package is uploaded. For the format of the payload, refer to the Decoding section.



Applications > sensecap-node > Devices > th-sensor > Data

Overview | Data | Settings

APPLICATION DATA

Filters: uplink | downlink | activation | ack | error

```
   time        counter   port
▲  19:25:48       4       2    retry confirmed   payload:  01 01 10 90 65 00 00 01 02 10 78 E6 00 00 92 AF
▼  19:25:47               0
▲  19:25:47       4       2    confirmed         payload:  01 01 10 90 65 00 00 01 02 10 78 E6 00 00 92 AF
▲  19:25:25       3       2                      payload:  01 06 00 00 00 00 00 2F 87
▼  19:25:05               0
▲  19:25:04       2       2    confirmed         payload:  01 06 00 00 00 00 00 2F 87
▼  19:24:48               0
▲  19:24:47       1       2    confirmed         payload:  01 06 00 00 00 00 00 2F 87
▼  19:24:30               0
▲  19:24:29       0       2    confirmed         payload:  00 00 00 03 03 00 02 00 07 00 4A 00 3C 00 01 01 00 00 01 00 01 01 02 00 99 00 30 12 01 03 00
⚡  19:24:19                                      dev addr:  26 01 27 DB   app eui:  80 00 00 00 00 00 00 06   dev eui:  2C F7 F1 20 14 70 02 97
```

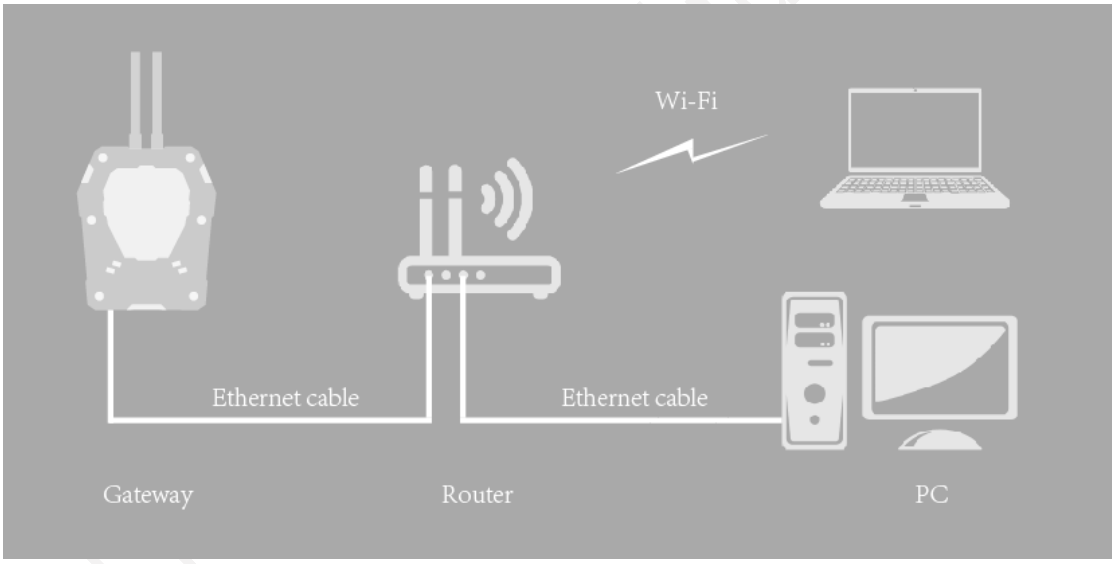
5 Add Gateway to ChirpStack LoRaWAN Network Server Stack

ChirpStack provides open-source components for LoRaWAN networks. Together they form a ready-to-use solution including an user-friendly web-interface for device management and APIs for integration.

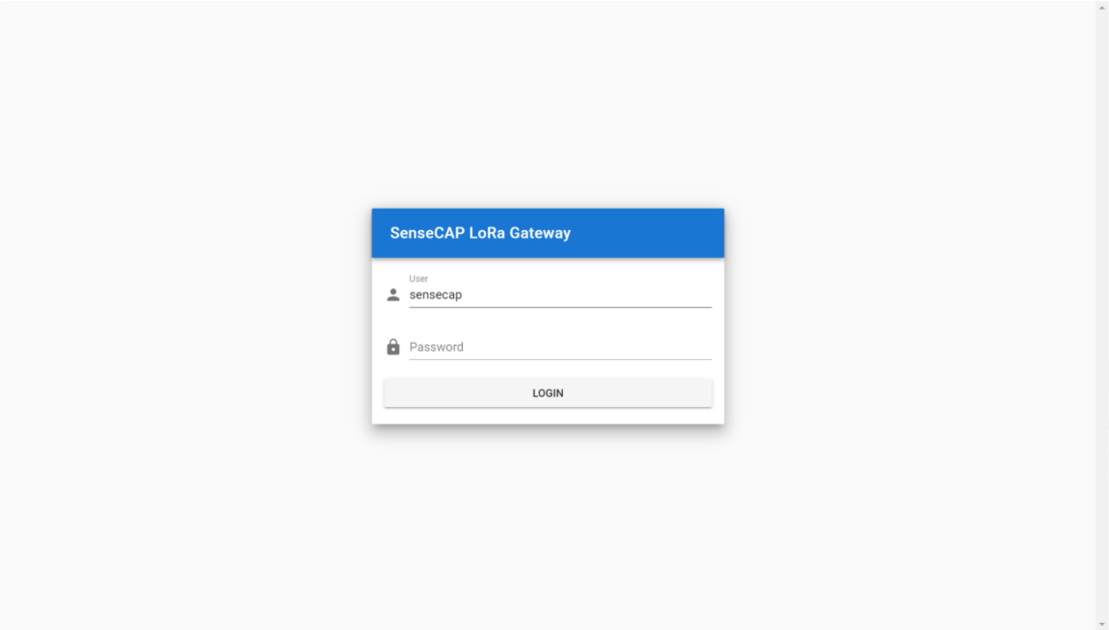
SenseCAP LoRaWAN Gateway has already integrated with ChirpStack LoRaWAN Network Server stack (hereinafter called the "ChirpStack LoRa Server"). The following LoRa Server components are accessible and configurable in Gateway: ChirpStack Gateway Bridge, ChirpStack Network Server and ChirpStack Application Server.

5.1 Turn on ChirpStack LoRa Server Mode

Prepare a router, and the network connection is shown in the figure:



(1) Check the IP of "sensecap" in the background of the router.
(2) Enter IP in the browser: IP:8000
If the IP is 192.168.1.1, enter 192.168.1.1:8000



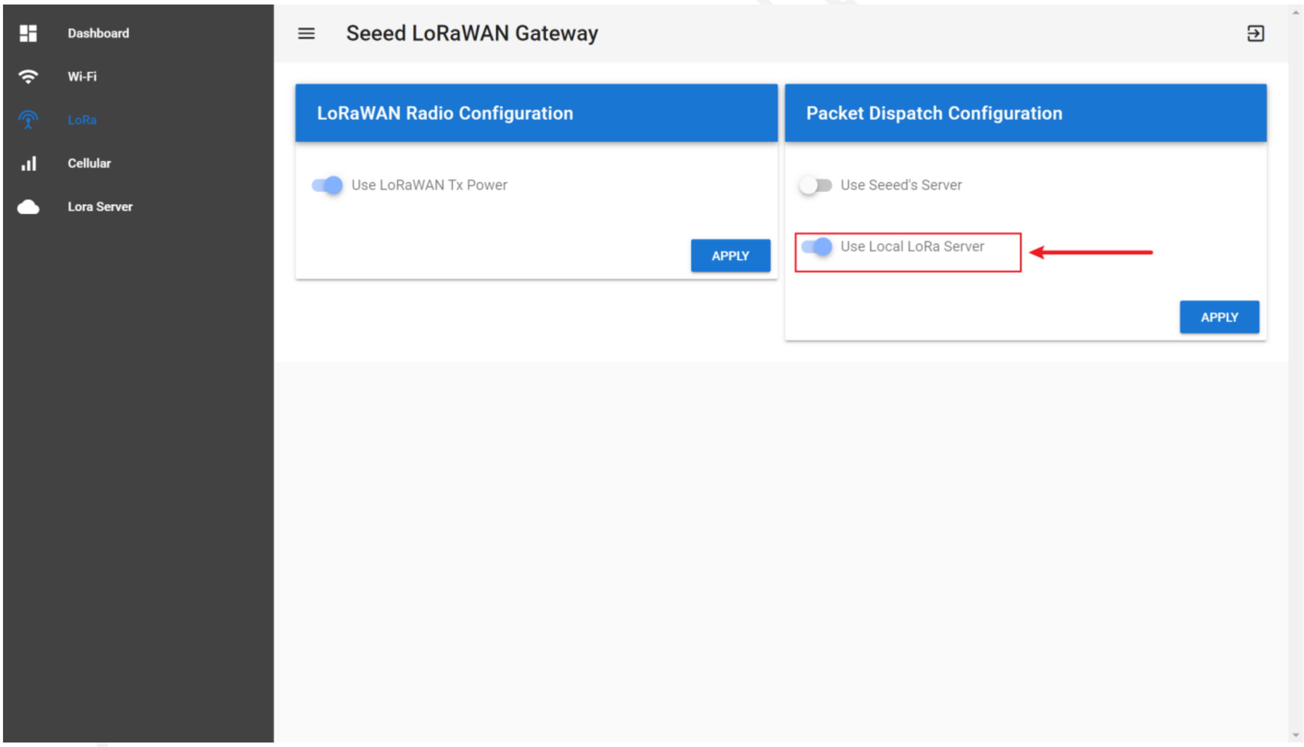
SenseCAP LoRa Gateway

User
sensecap

Password

(3) User: sensecap
Password: sensecap!!!

(4) Turn off the “Use Seeed’s Server”, and turn on “Use Local LoRa Server”.



Seeed LoRaWAN Gateway

LoRaWAN Radio Configuration

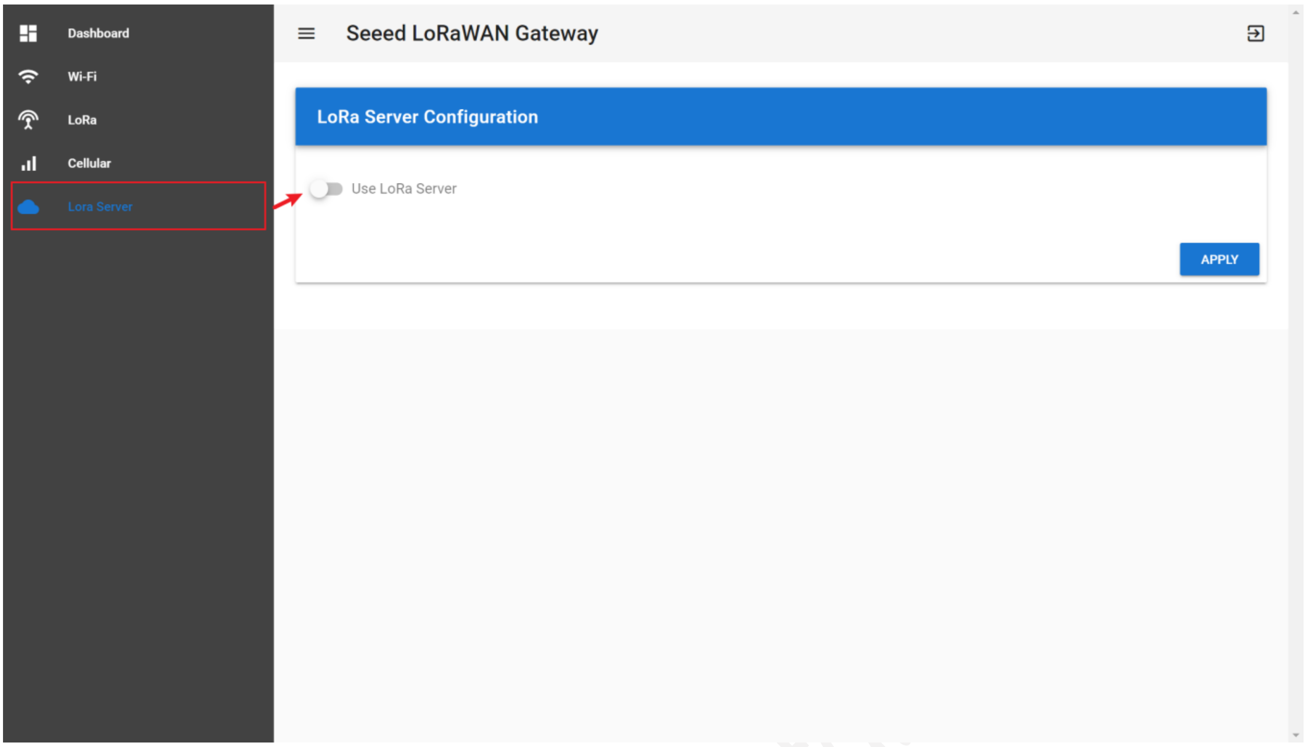
● Use LoRaWAN Tx Power

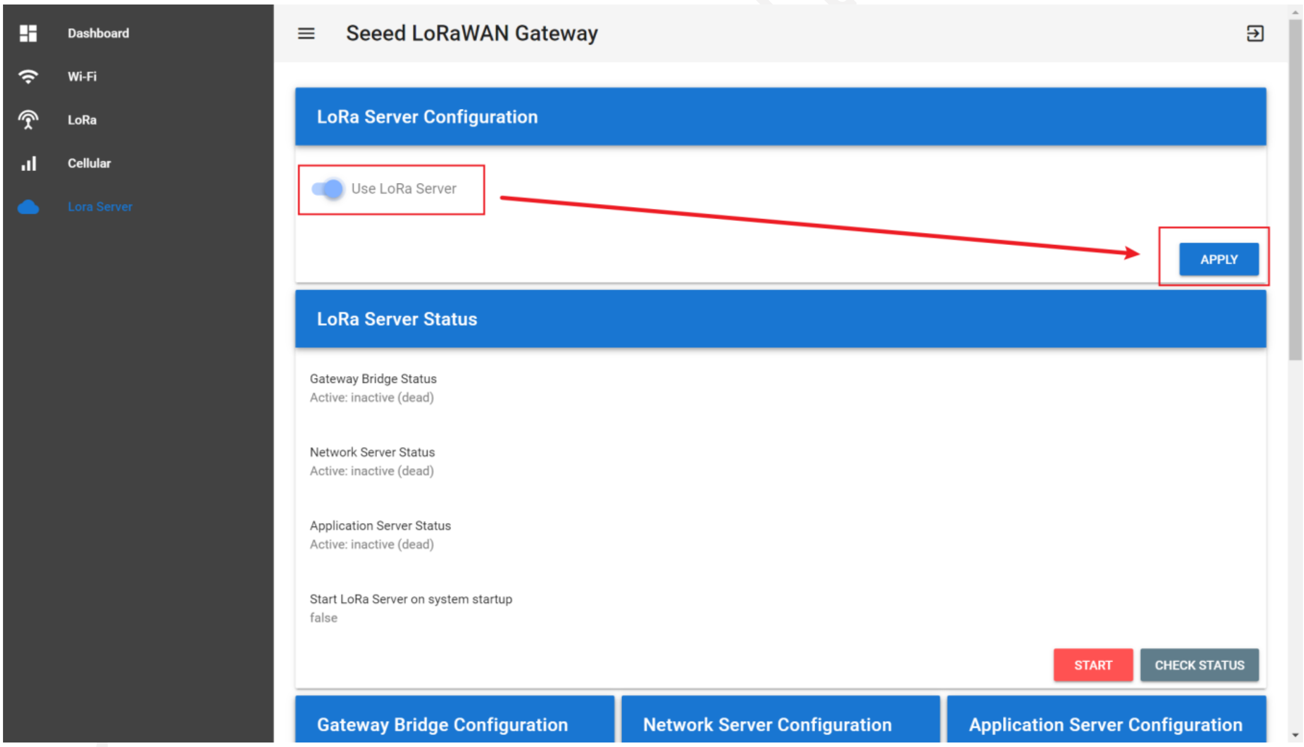
Packet Dispatch Configuration

○ Use Seeed's Server

● Use Local LoRa Server

(5) Turn on the “Use LoRa Server” button, and apply. (“LoRa Server” is the name of ChirpStack LoRa Server)




Dashboard
Wi-Fi
LoRa
Cellular
Lora Server
Seeed LoRaWAN Gateway
LoRa Server Configuration
Use LoRa Server
APPLY
LoRa Server Status
START
CHECK STATUS
Gateway Bridge Configuration
Network Server Configuration
Application Server Configuration

Gateway Bridge Status
Active: inactive (dead)

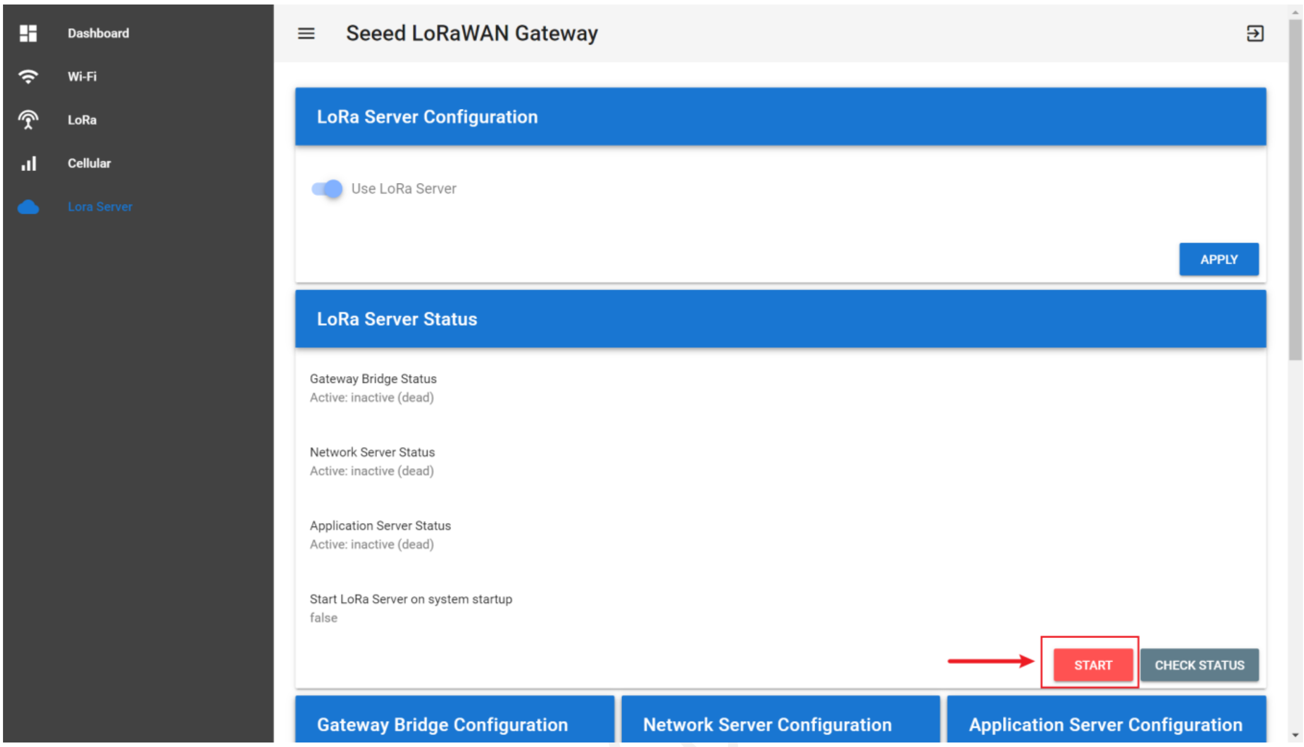
Network Server Status
Active: inactive (dead)

Application Server Status
Active: inactive (dead)

Start LoRa Server on system startup
false

5.2 ChirpStack LoRa Server Configuration

First, click the “Start” button to start the service.


Dashboard
Wi-Fi
LoRa
Cellular
Lora Server
Seeed LoRaWAN Gateway
LoRa Server Configuration
Use LoRa Server
APPLY
LoRa Server Status
START
CHECK STATUS
Gateway Bridge Configuration
Network Server Configuration
Application Server Configuration

Gateway Bridge Status
Active: inactive (dead)

Network Server Status
Active: inactive (dead)

Application Server Status
Active: inactive (dead)

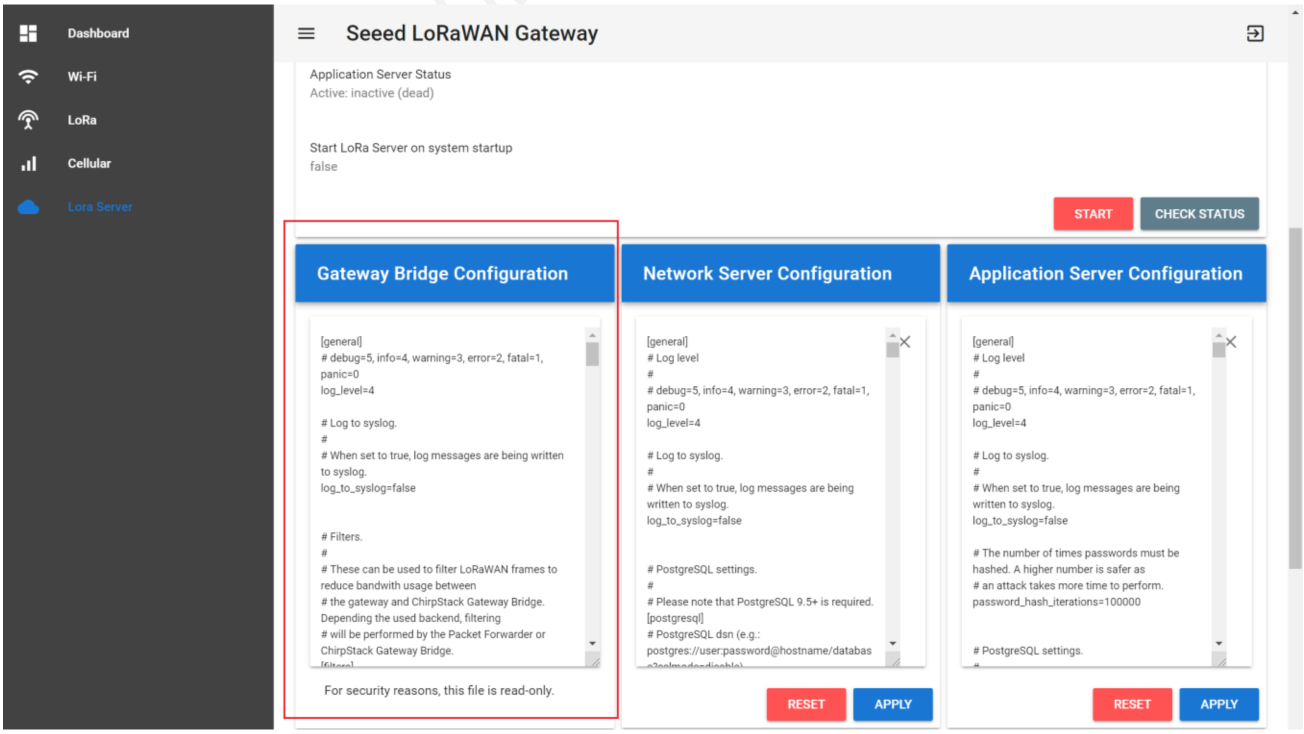
Start LoRa Server on system startup
false

(1) ChirpStack Gateway Bridge:

Refer to: https://www.chirpstack.io/gateway-bridge/

It converts LoRa® Packet Forwarder protocols into a ChirpStack Network Server common data-format (JSON and Protobuf).

For security reasons, this file is read-only.



Application Server Status
Active: inactive (dead)

Start LoRa Server on system startup
false

Gateway Bridge Configuration

```
[general]
# debug=5, info=4, warning=3, error=2, fatal=1,
panic=0
log_level=4

# Log to syslog.
#
# When set to true, log messages are being written
to syslog.
log_to_syslog=false


# Filters.
#
# These can be used to filter LoRaWAN frames to
reduce bandwith usage between
# the gateway and ChirpStack Gateway Bridge.
Depending the used backend, filtering
# will be performed by the Packet Forwarder or
ChirpStack Gateway Bridge.
[filters]
```

For security reasons, this file is read-only.

Network Server Configuration

```
[general]
# Log level
#
# debug=5, info=4, warning=3, error=2, fatal=1,
panic=0
log_level=4

# Log to syslog.
#
# When set to true, log messages are being
written to syslog.
log_to_syslog=false


# PostgreSQL settings.
#
# Please note that PostgreSQL 9.5+ is required.
[postgresql]
# PostgreSQL dsn (e.g.:
postgres://user:password@hostname/databas
e?sslmode=disable).
```

Application Server Configuration

```
[general]
# Log level
#
# debug=5, info=4, warning=3, error=2, fatal=1,
panic=0
log_level=4

# Log to syslog.
#
# When set to true, log messages are being
written to syslog.
log_to_syslog=false

# The number of times passwords must be
hashed. A higher number is safer as
# an attack takes more time to perform.
password_hash_iterations=100000


# PostgreSQL settings.
```

(2) ChirpStack Network Server:

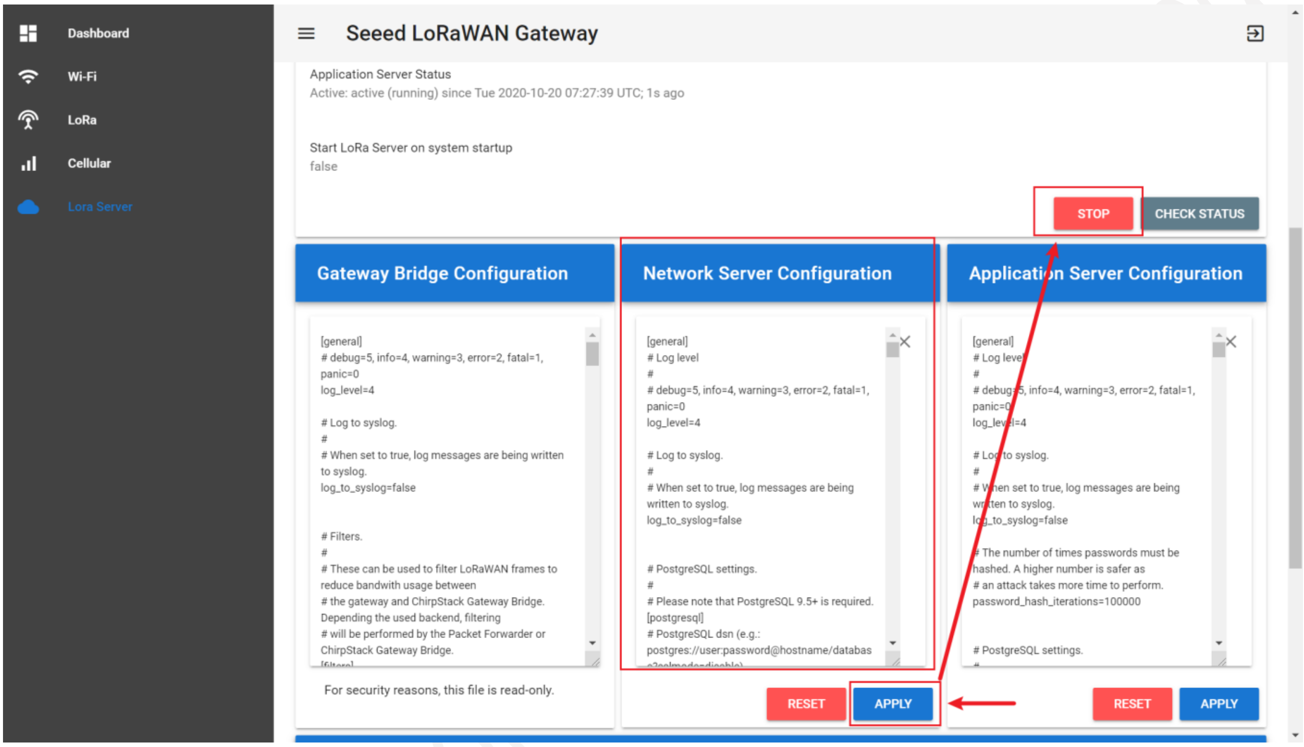
Refer to: https://www.chirpstack.io/network-server/

The responsibility of the Network Server component is the de-duplication of received LoRaWAN frames by the LoRa® gateways and for the collected frames handle the: Authentication; LoRaWAN mac-layer (and mac-commands); Communication with the ChirpStack Application Server; Scheduling of downlink frames.

In general, the default configuration is used. Please refer to the official tutorial before making any modifications.

Click "APPLY" to save the configuration after making changes.

Then, click "STOP" in "Application Server Status" and finally click "START" to make the configuration take effect.



Application Server Status
Active: active (running) since Tue 2020-10-20 07:27:39 UTC; 1s ago

Start LoRa Server on system startup
false

Gateway Bridge Configuration

```
[general]
# debug=5, info=4, warning=3, error=2, fatal=1,
panic=0
log_level=4

# Log to syslog.
#
# When set to true, log messages are being written
to syslog.
log_to_syslog=false


# Filters.
#
# These can be used to filter LoRaWAN frames to
reduce bandwith usage between
# the gateway and ChirpStack Gateway Bridge.
Depending the used backend, filtering
# will be performed by the Packet Forwarder or
ChirpStack Gateway Bridge.
[filters]
```

For security reasons, this file is read-only.

Network Server Configuration

```
[general]
# Log level
#
# debug=5, info=4, warning=3, error=2, fatal=1,
panic=0
log_level=4

# Log to syslog.
#
# When set to true, log messages are being
written to syslog.
log_to_syslog=false


# PostgreSQL settings.
#
# Please note that PostgreSQL 9.5+ is required.
[postgresql]
# PostgreSQL dsn (e.g.:
postgres://user:password@hostname/databas
e?sslmode=disable).
```

Application Server Configuration

```
[general]
# Log level
#
# debug=5, info=4, warning=3, error=2, fatal=1,
panic=0
log_level=4

# Log to syslog.
#
# When set to true, log messages are being
written to syslog.
log_to_syslog=false

# The number of times passwords must be
hashed. A higher number is safer as
# an attack takes more time to perform.
password_hash_iterations=100000


# PostgreSQL settings.
#
```

(3) ChirpStack Application Server:

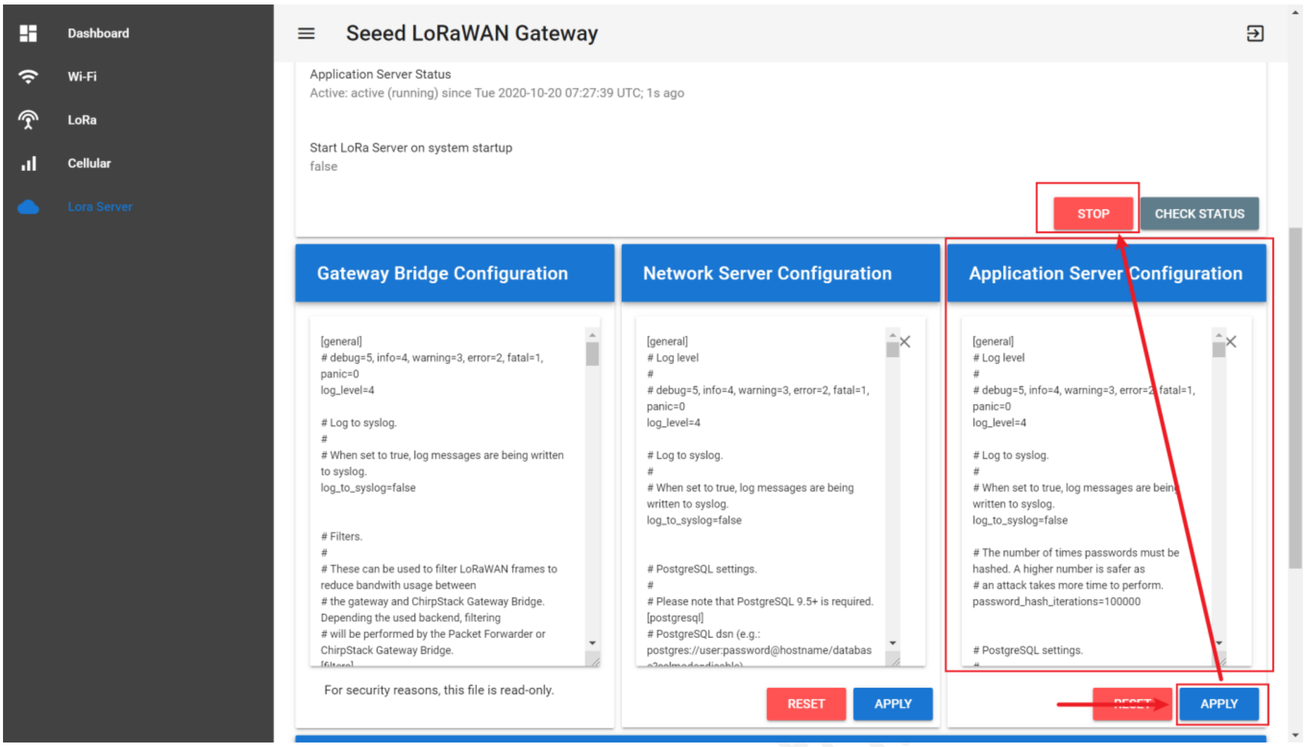
Refer to: https://www.chirpstack.io/application-server/

It is responsible for the device “inventory” part of a LoRaWAN infrastructure, handling of join-request and the handling and encryption of application payloads.

In general, the default configuration is used. Please refer to the official tutorial before making any modifications.

Click "APPLY" to save the configuration after making changes.

Then, click "STOP" in "Application Server Status" and finally click "START" to make the configuration take effect.



Application Server Status
Active: active (running) since Tue 2020-10-20 07:27:39 UTC; 1s ago

Start LoRa Server on system startup
false

Gateway Bridge Configuration

```
[general]
# debug=5, info=4, warning=3, error=2, fatal=1,
panic=0
log_level=4

# Log to syslog.
#
# When set to true, log messages are being written
to syslog.
log_to_syslog=false


# Filters.
#
# These can be used to filter LoRaWAN frames to
reduce bandwith usage between
# the gateway and ChirpStack Gateway Bridge.
Depending the used backend, filtering
# will be performed by the Packet Forwarder or
ChirpStack Gateway Bridge.
```

For security reasons, this file is read-only.

Network Server Configuration

```
[general]
# Log level
#
# debug=5, info=4, warning=3, error=2, fatal=1,
panic=0
log_level=4

# Log to syslog.
#
# When set to true, log messages are being
written to syslog.
log_to_syslog=false


# PostgreSQL settings.
#
# Please note that PostgreSQL 9.5+ is required.
[postgresql]
# PostgreSQL dsn (e.g.:
postgres://user:password@hostname/databas
```

Application Server Configuration

```
[general]
# Log level
#
# debug=5, info=4, warning=3, error=2, fatal=1,
panic=0
log_level=4

# Log to syslog.
#
# When set to true, log messages are being
written to syslog.
log_to_syslog=false

# The number of times passwords must be
hashed. A higher number is safer as
# an attack takes more time to perform.
password_hash_iterations=100000


# PostgreSQL settings.
```

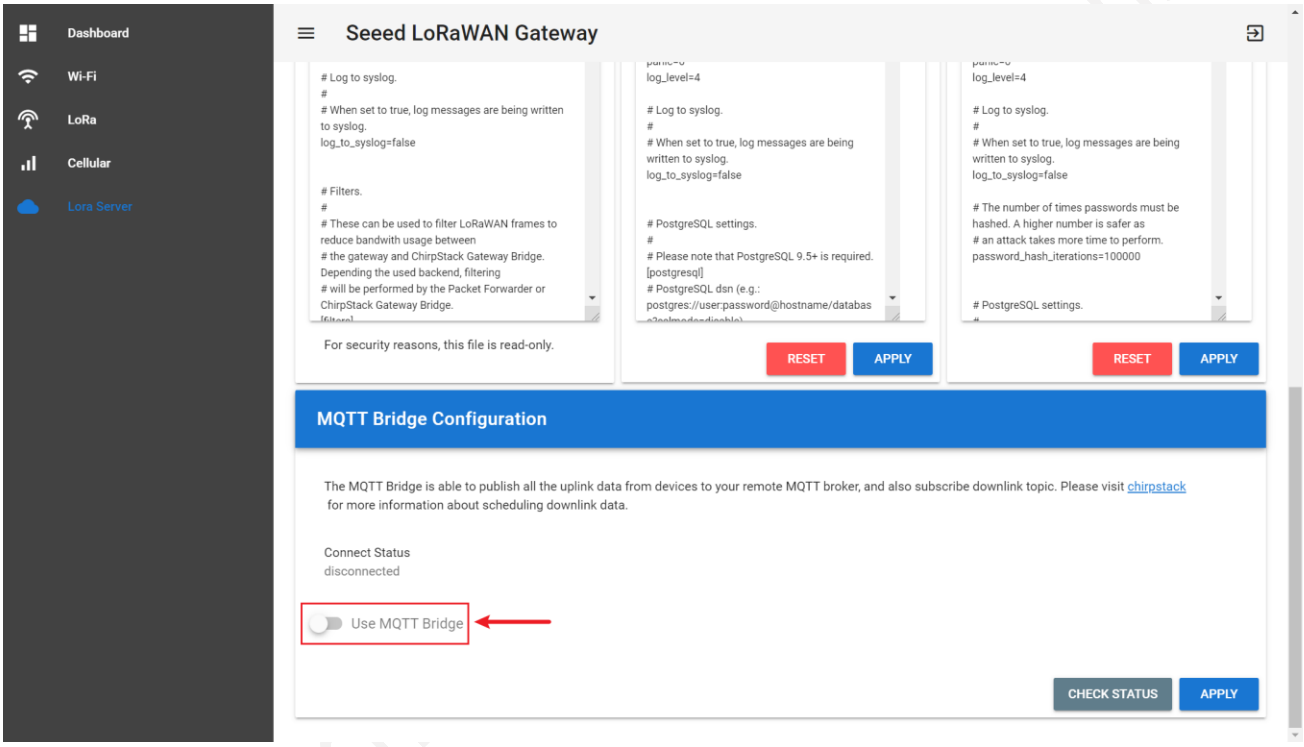
(4) If you have the wrong configuration, click "RESET" to restore the default configuration.

5.3 MQTT Bridge Configuration

The MQTT Bridge is able to publish all the uplink data from devices to your remote MQTT broker, and also subscribe downlink topic. Please visit ChirpStack(https://www.chirpstack.io/application-server/integrations/mqtt/) for more information about scheduling downlink data.

5.3.1 Gateway Configuration

(1) Click "Use MQTT Bridge".



```
# Log to syslog.
#
# When set to true, log messages are being written to syslog.
log_to_syslog=false


# Filters.
#
# These can be used to filter LoRaWAN frames to reduce bandwith usage between
# the gateway and ChirpStack Gateway Bridge. Depending the used backend, filtering
# will be performed by the Packet Forwarder or ChirpStack Gateway Bridge.
```

For security reasons, this file is read-only.

```
log_level=4

# Log to syslog.
#
# When set to true, log messages are being written to syslog.
log_to_syslog=false


# PostgreSQL settings.
#
# Please note that PostgreSQL 9.5+ is required.
[postgresql]
# PostgreSQL dsn (e.g.: postgres://user:password@hostname/databas
```

```
log_level=4

# Log to syslog.
#
# When set to true, log messages are being written to syslog.
log_to_syslog=false

# The number of times passwords must be hashed. A higher number is safer as
# an attack takes more time to perform.
password_hash_iterations=100000


# PostgreSQL settings.
```

MQTT Bridge Configuration

The MQTT Bridge is able to publish all the uplink data from devices to your remote MQTT broker, and also subscribe downlink topic. Please visit chirpstack for more information about scheduling downlink data.

Connect Status
disconnected

○ Use MQTT Bridge

(2) After filling in each parameter, click "APPLY".

①

MQTT Server address: mqtt://xxx.xx or mqtts://xxx.xx

If xxx.xx (IP) is 111.230.200.102, the address is mqtt://111.230.200.102 or mqtts://111.230.200.102
If xxx.xx (url) is mybroker.com, the address is mqtt:// mybroker.com or mqtts:// mybroker.com

②

MQTT Server 's Port.
In general, mqtt corresponds to port 1883 and mqtts to port 8883.

③

Keepalive:

60 is default value. When the MQTT connection between the Gateway and the Server is disconnected over 60 seconds, it determines that the client is offline.

0 means turn off the keepalive function.

④

CleanSession:

true: the gateway reconnects to the network after a power outage or disconnection, and cannot receive data from MQTTpub to the gateway for that period.

false: the gateway reconnects to the network after a power outage or disconnection, and can receive data from MQTTpub to the gateway for that period.

⑤

Username: Null if none, depending on the server configuration.

⑥

Password: Null if none, depending on the server configuration.

⑦

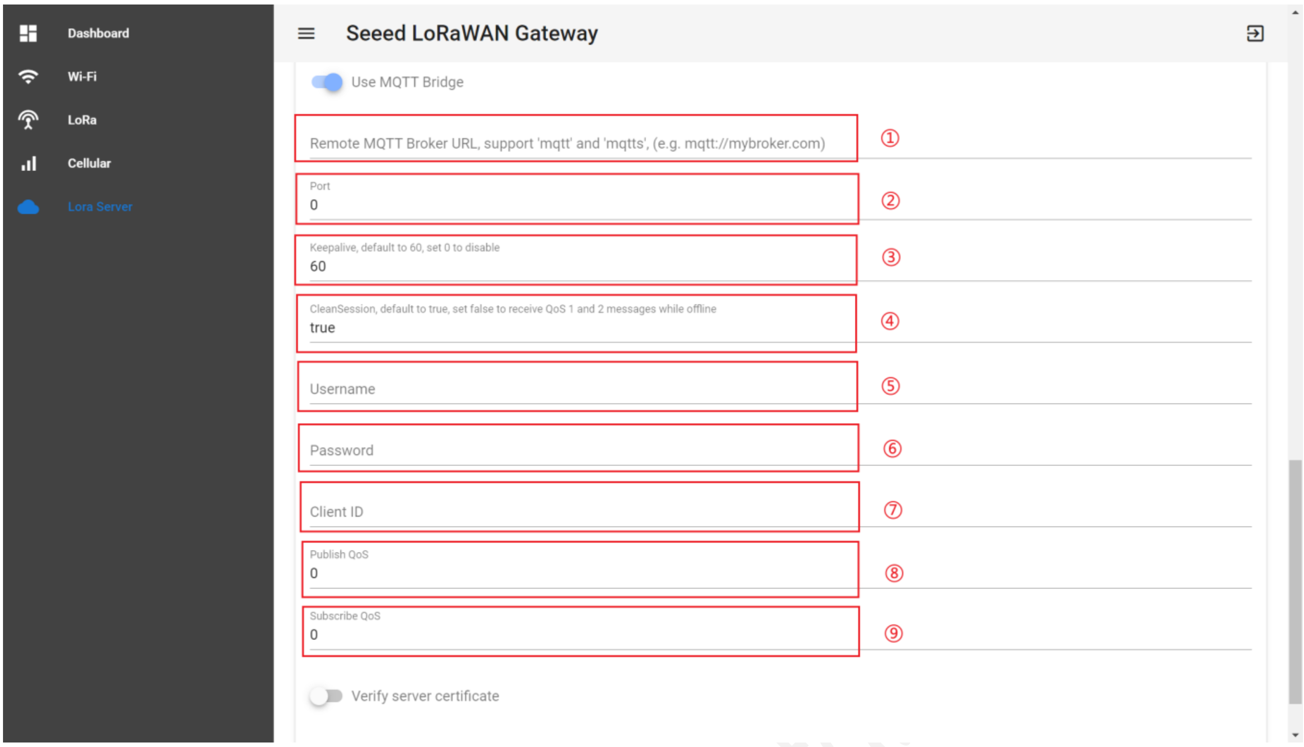
Client ID: Custom the name, and each Client ID is unique to the same MQTT server.

⑧

Publish QoS: 0, 1 or 2. (refer to the MQTT rules)

⑨

Subscribe QoS: 0, 1 or 2. (refer to the MQTT rules)



Use MQTT Bridge

Remote MQTT Broker URL, support 'mqtt' and 'mqtts', (e.g. mqtt://mybroker.com)

Port: 0

Keepalive, default to 60, set 0 to disable: 60

CleanSession, default to true, set false to receive QoS 1 and 2 messages while offline: true

Username

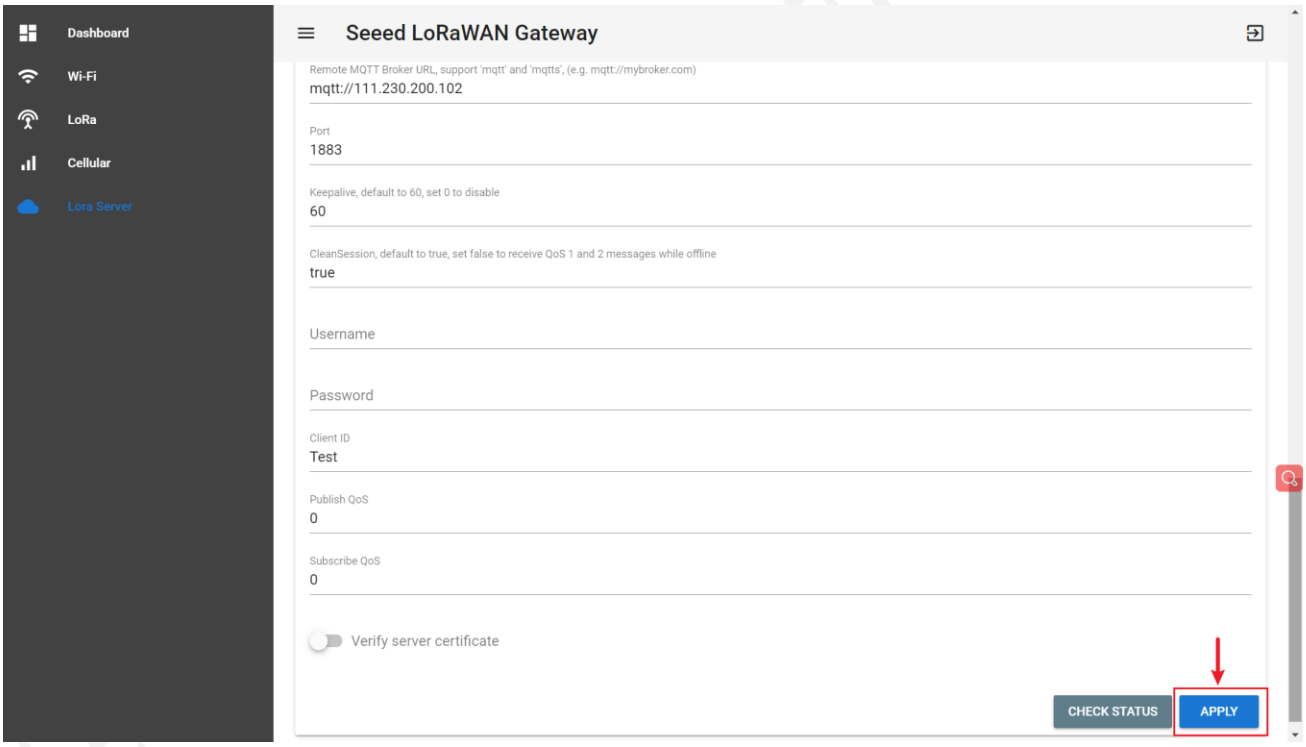
Password

Client ID

Publish QoS: 0

Subscribe QoS: 0

Verify server certificate



Remote MQTT Broker URL, support 'mqtt' and 'mqtts', (e.g. mqtt://mybroker.com): mqtt://111.230.200.102

Port: 1883

Keepalive, default to 60, set 0 to disable: 60

CleanSession, default to true, set false to receive QoS 1 and 2 messages while offline: true

Username

Password

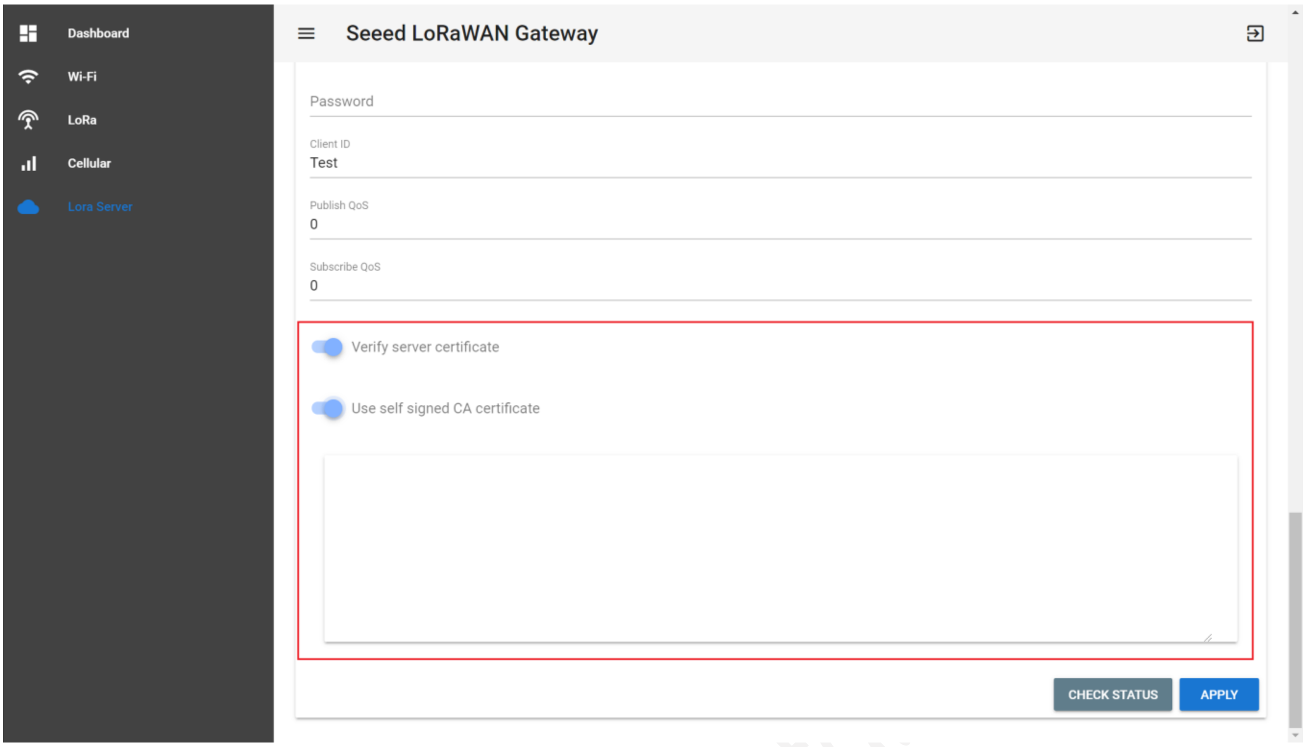
Client ID: Test

Publish QoS: 0

Subscribe QoS: 0

Verify server certificate

(3) It is off by default and can generally be ignored: Verify server certificate.
If true, the server certificate is verified against the list of supplied CAs.
If false, the server certificate is verified against your self-signed certificate.



Password

Client ID
Test

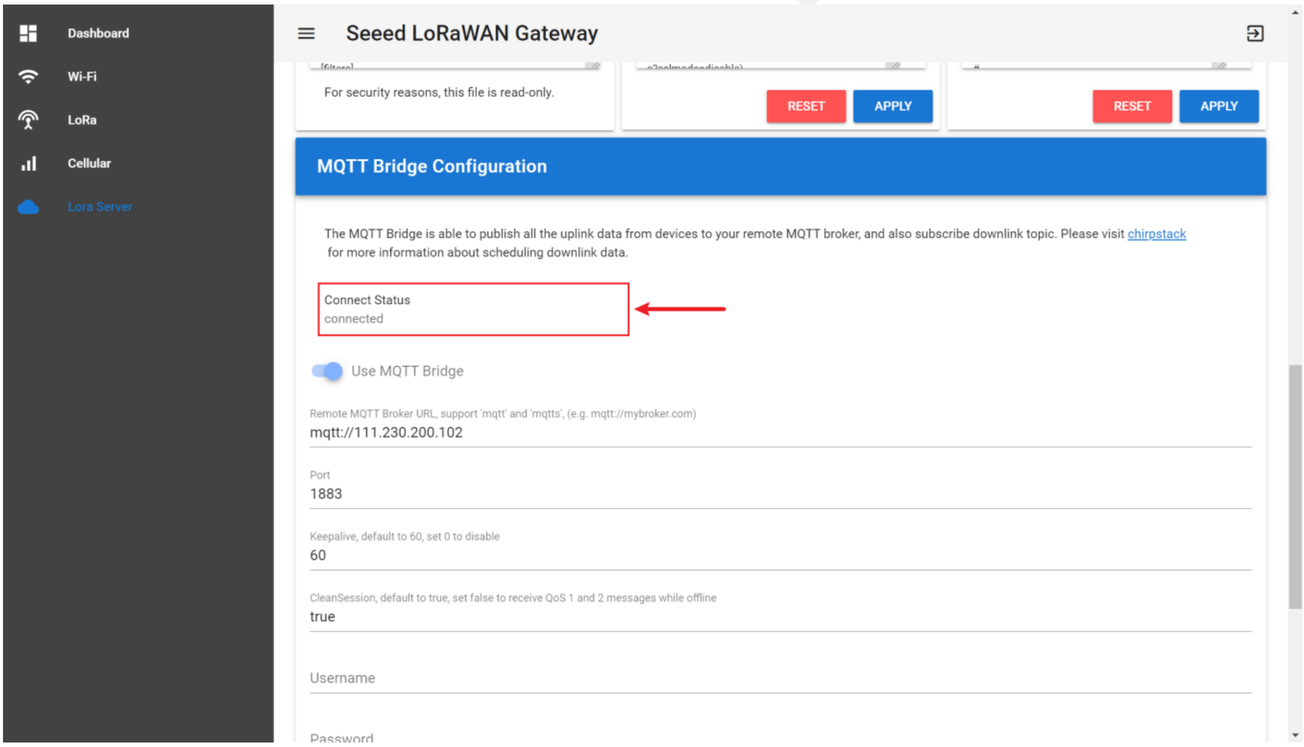
Publish QoS
0

Subscribe QoS
0

Verify server certificate

Use self signed CA certificate

(4) Check Status: Disconnected / Reconnecting / Connected.



For security reasons, this file is read-only.

The MQTT Bridge is able to publish all the uplink data from devices to your remote MQTT broker, and also subscribe downlink topic. Please visit chirpstack for more information about scheduling downlink data.

Connect Status
connected

Use MQTT Bridge

Remote MQTT Broker URL, support 'mqtt' and 'mqtts', (e.g. mqtt://mybroker.com)
mqtt://111.230.200.102

Port
1883

Keepalive, default to 60, set 0 to disable
60

CleanSession, default to true, set false to receive QoS 1 and 2 messages while offline
true

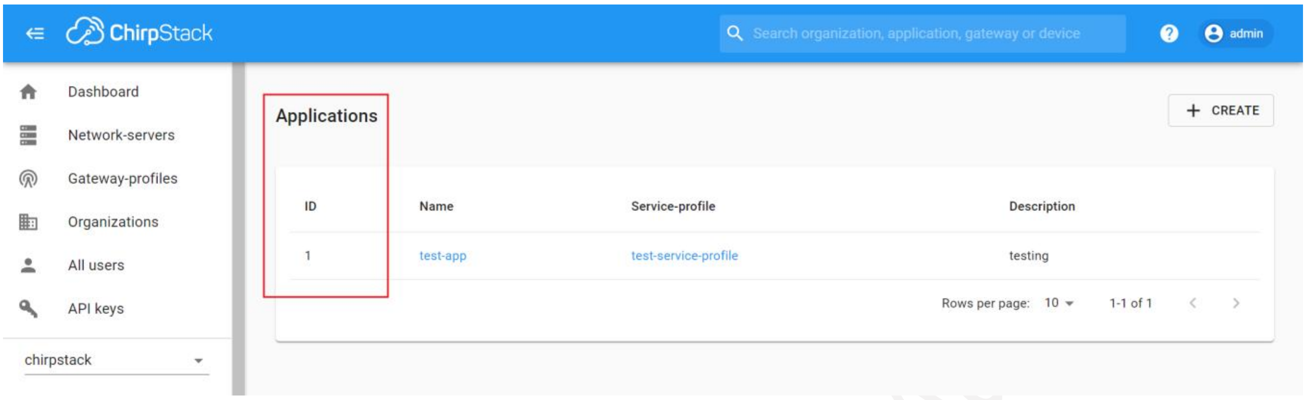
Username

Password

5.3.2 MQTT Client Configuration

For details, please refer to: https://www.chirpstack.io/application-server/integrations/events/#ack

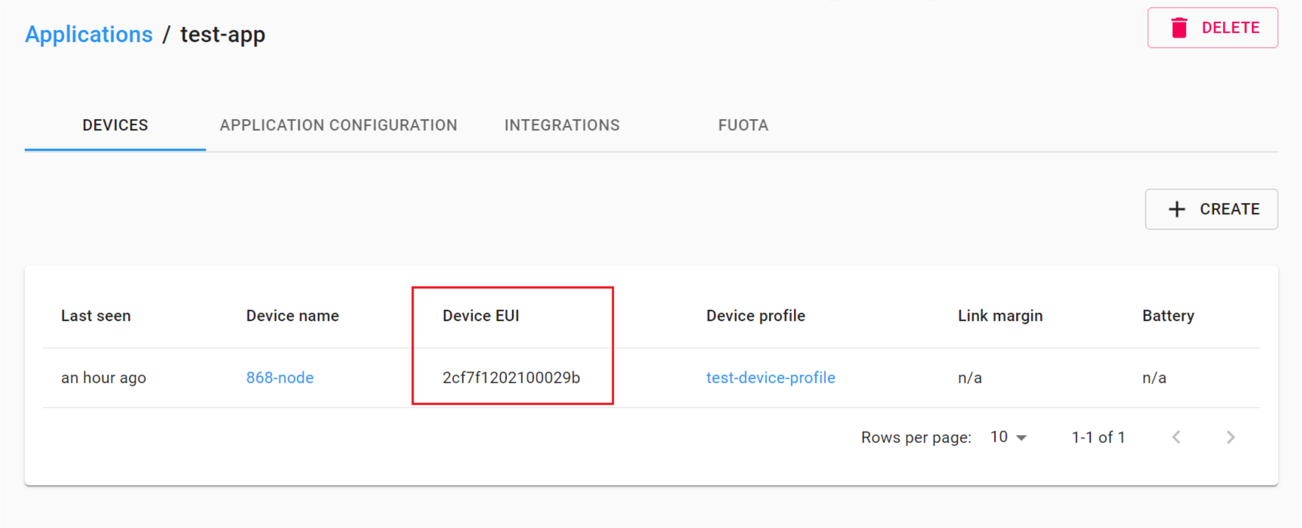
ApplicationID: the Application ID.



ID	Name	Service-profile	Description
1	test-app	test-service-profile	testing

Rows per page: 10 1-1 of 1

DevEUI: Device EUI.



Last seen	Device name	Device EUI	Device profile	Link margin	Battery
an hour ago	868-node	2cf7f1202100029b	test-device-profile	n/a	n/a

Rows per page: 10 1-1 of 1

(1) Device data subscription

```
application/[ApplicationID]/device/[DevEUI]/event/up

e.g. application/1/device/ 2cf7f1202100029b/event/up
```

(2) Join packet subscription

```
application/[ApplicationID]/device/[DevEUI]/event/join

e.g. application/1/device/ 2cf7f1202100029b/event/join
```

(3) Status packet subscription

```
application/[ApplicationID]/device/[DevEUI]/event/status

e.g. application/1/device/ 2cf7f1202100029b/event/ status
```

5.3.3 Scheduling a Downlink

The default topic for scheduling downlink payloads is:

```
application/[ApplicationID]/device/[DevEUI]/command/down
```

The ApplicationID and DevEUI of the device will be taken from the topic.

Example payload:

```
{
        "confirmed": true,          // whether the payload must be sent as confirmed data down or not
        "fPort": 10,                // FPort to use (must be > 0)
        "data": "...."              // base64 encoded data (plaintext, will be encrypted by ChirpStack Network Server)
        "object": {                 // decoded object (when application coded has been configured)
                "temperatureSensor": {"1": 25},      // when providing the 'object', you can omit 'data'
                "humiditySensor": {"1": 32}
        }
}
```

5.4 ChirpStack Application Server

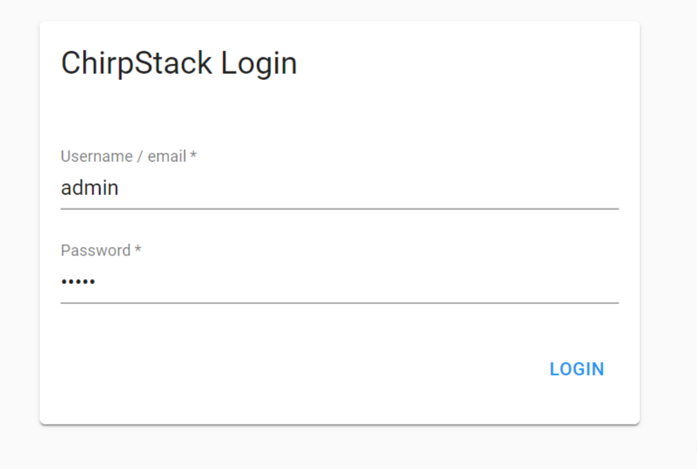
5.4.1 Log on to the background

According to the Gateway IP obtained in Section 4.1, log in the Web UI.

The login address: IP:8080 (if IP is 192.168.8.100, enter 192.168.8.100:8080)

Username(default): admin

Password(default): admin

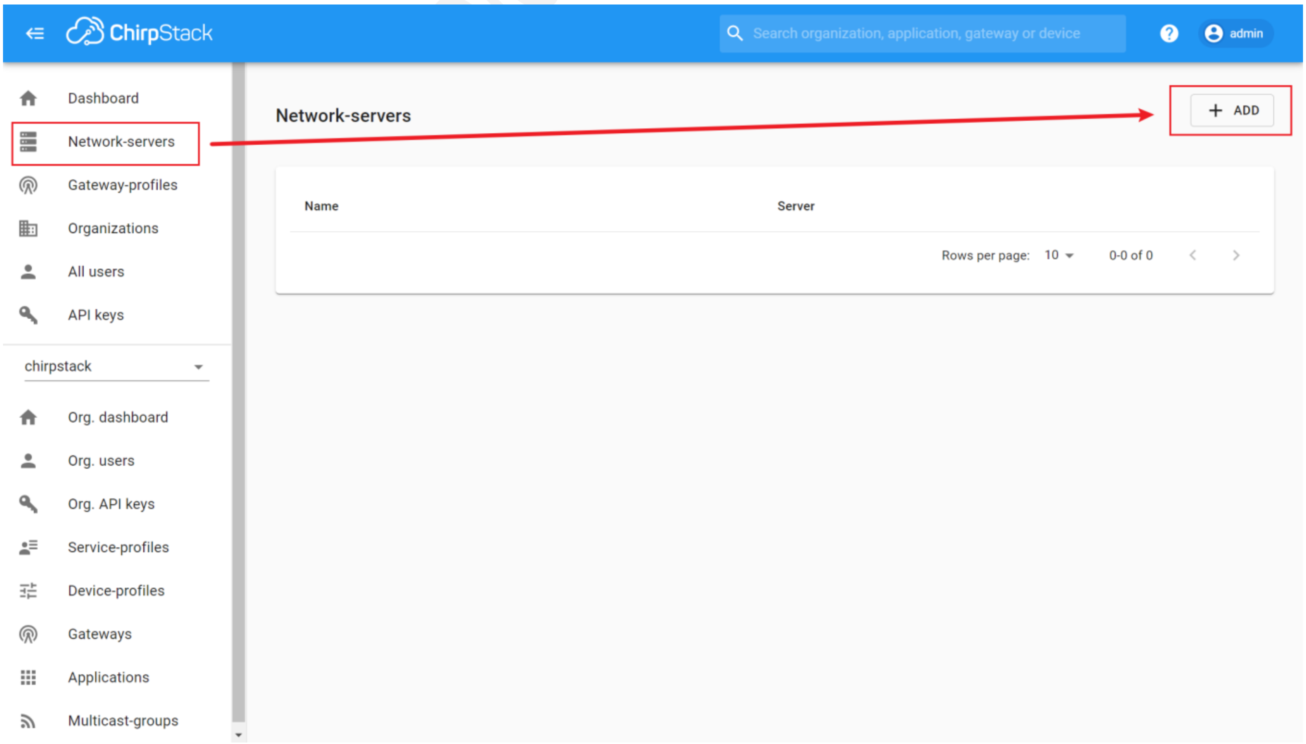


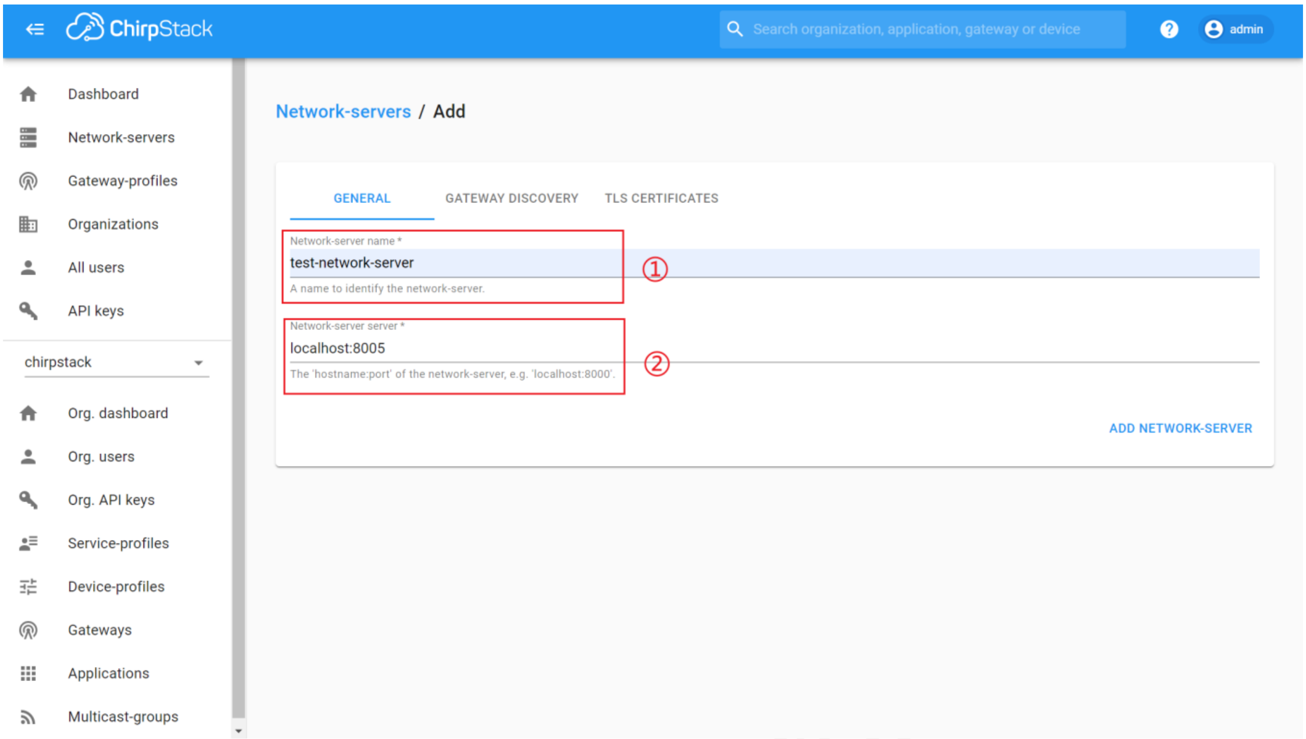
ChirpStack Login

Username / email *
admin

Password *
•••••

5.4.2 Add the Network-servers





Network-servers / Add

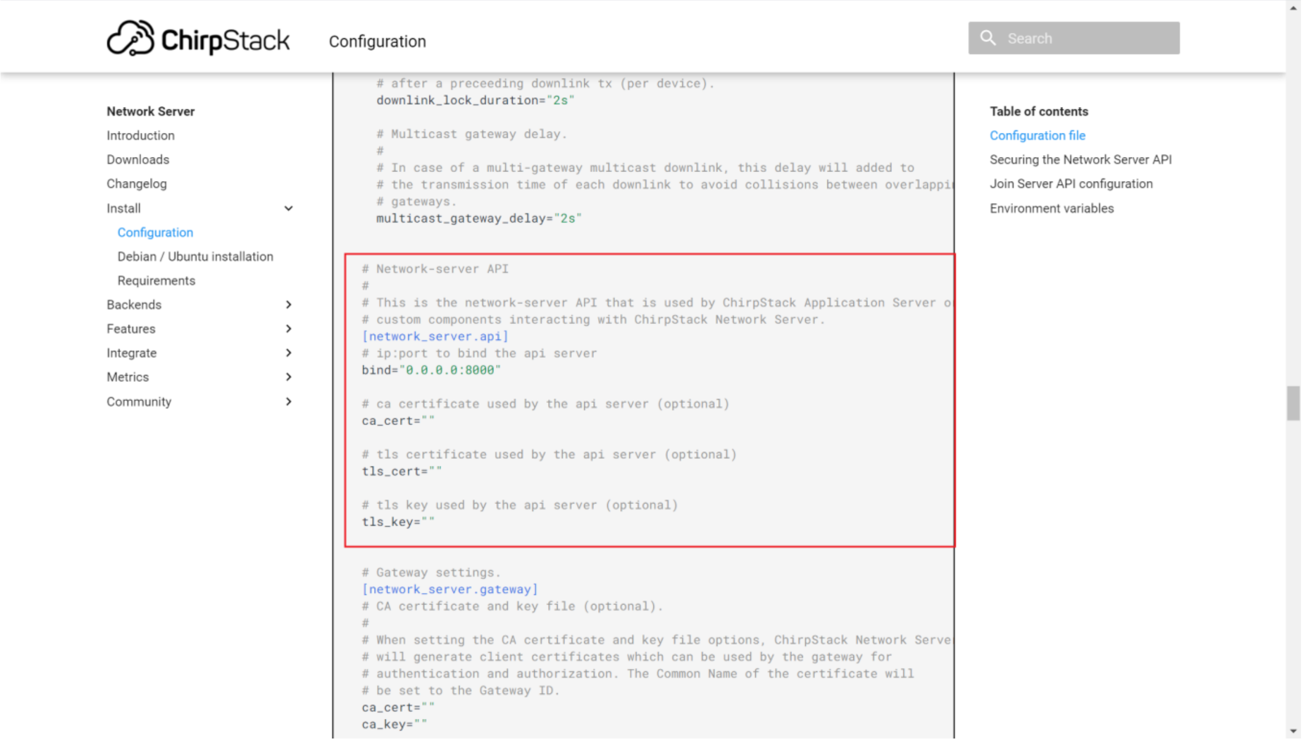
Network-server name *
test-network-server
A name to identify the network-server.

Network-server server *
localhost:8005
The 'hostname:port' of the network-server, e.g. 'localhost:8000'.

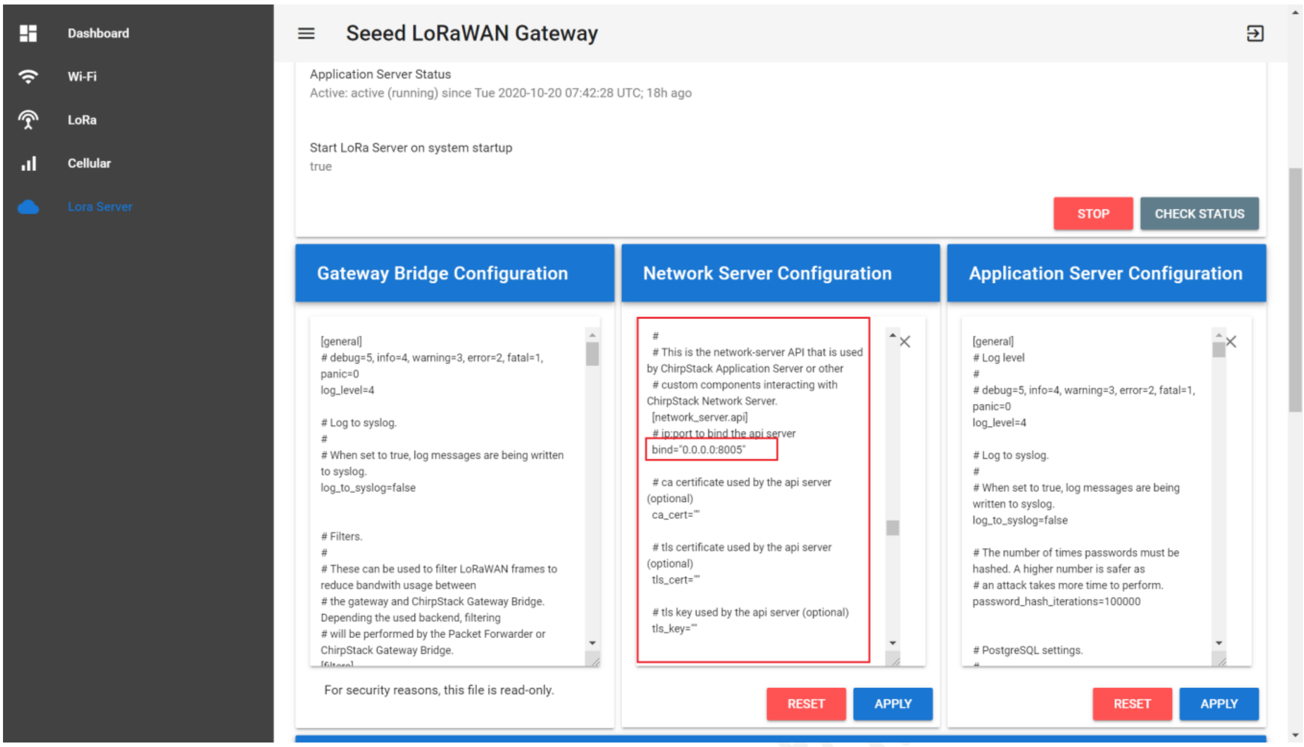
① Network-server name: custom name.

② Network-server server: the default value is localhost:8005

Refer to: https://www.chirpstack.io/network-server/install/config/ . You can modify it in the “Network Server Configuration”.



```
  # after a preceeding downlink tx (per device).
  downlink_lock_duration="2s"

  # Multicast gateway delay.
  #
  # In case of a multi-gateway multicast downlink, this delay will added to
  # the transmission time of each downlink to avoid collisions between overlappi
  # gateways.
  multicast_gateway_delay="2s"

# Network-server API
#
# This is the network-server API that is used by ChirpStack Application Server o
# custom components interacting with ChirpStack Network Server.
[network_server.api]
# ip:port to bind the api server
bind="0.0.0.0:8000"

# ca certificate used by the api server (optional)
ca_cert=""

# tls certificate used by the api server (optional)
tls_cert=""

# tls key used by the api server (optional)
tls_key=""

# Gateway settings.
[network_server.gateway]
# CA certificate and key file (optional).
#
# When setting the CA certificate and key file options, ChirpStack Network Serve
# will generate client certificates which can be used by the gateway for
# authentication and authorization. The Common Name of the certificate will
# be set to the Gateway ID.
ca_cert=""
ca_key=""
```



Application Server Status
Active: active (running) since Tue 2020-10-20 07:42:28 UTC; 18h ago

Start LoRa Server on system startup
true

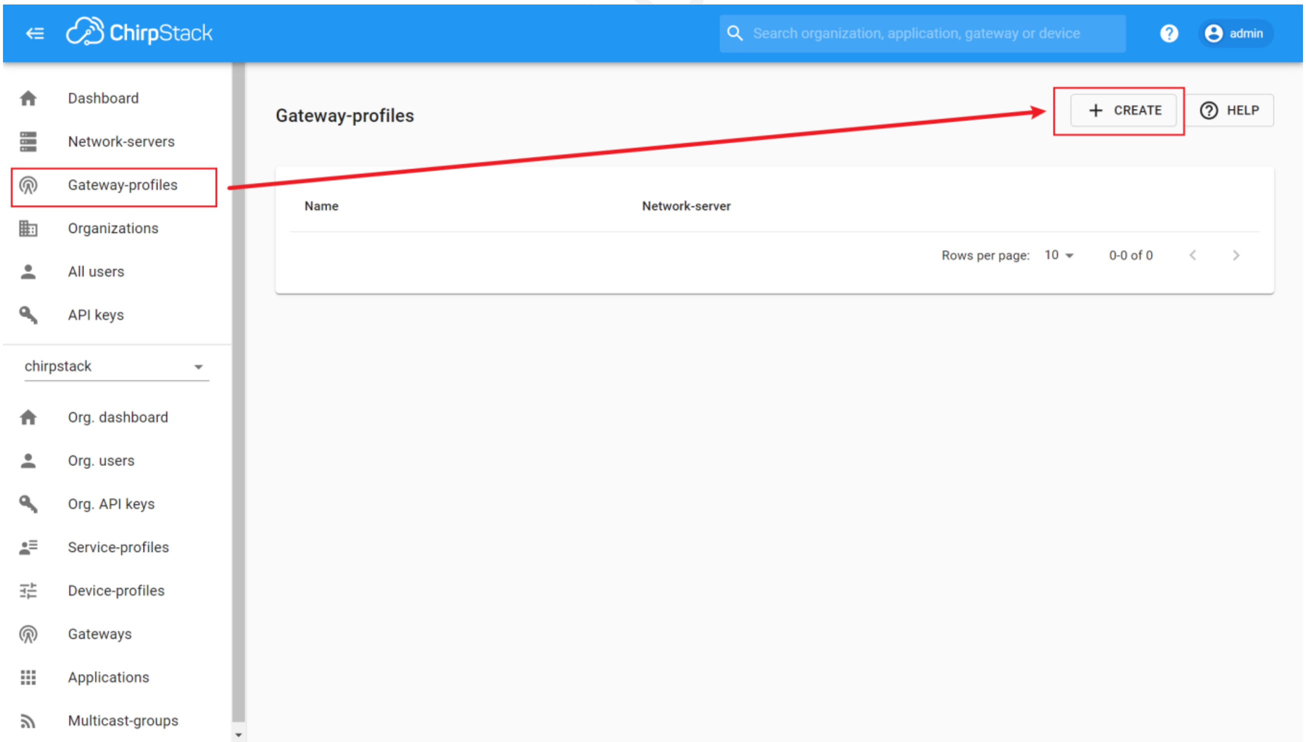
```
[general]
# debug=5, info=4, warning=3, error=2, fatal=1, panic=0
log_level=4

# Log to syslog.
#
# When set to true, log messages are being written to syslog.
log_to_syslog=false


# Filters.
#
# These can be used to filter LoRaWAN frames to reduce bandwith usage between
# the gateway and ChirpStack Gateway Bridge. Depending the used backend, filtering
# will be performed by the Packet Forwarder or ChirpStack Gateway Bridge.
[filters]
```

For security reasons, this file is read-only.

```
 #
 # This is the network-server API that is used by ChirpStack Application Server or other
 # custom components interacting with ChirpStack Network Server.
 [network_server.api]
 # ip:port to bind the api server
 bind="0.0.0.0:8005"

 # ca certificate used by the api server (optional)
 ca_cert=""

 # tls certificate used by the api server (optional)
 tls_cert=""

 # tls key used by the api server (optional)
 tls_key=""
```

```
[general]
# Log level
#
# debug=5, info=4, warning=3, error=2, fatal=1, panic=0
log_level=4

# Log to syslog.
#
# When set to true, log messages are being written to syslog.
log_to_syslog=false

# The number of times passwords must be hashed. A higher number is safer as
# an attack takes more time to perform.
password_hash_iterations=100000


# PostgreSQL settings.
#
```

5.4.3 Create the Gateway-profiles



Name	Network-server

Rows per page: 10 0-0 of 0

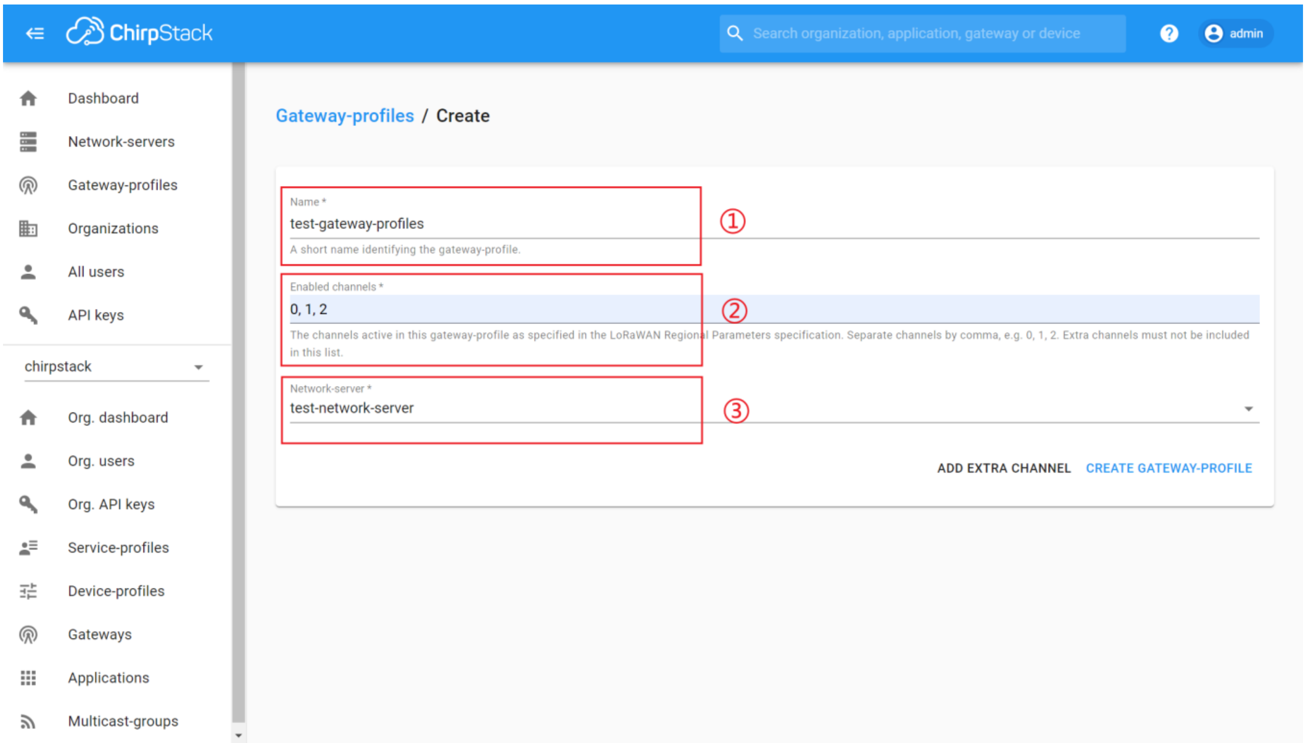
① Name: custom name.

② Enabled channels: 0, 1, 2

EU channels: 0, 1, 2

US902-923 channels (sub-band 2): 8, 9, 10, 11, 12, 13, 14, 15, 65

③ Network-server: select the Network-server you created earlier.

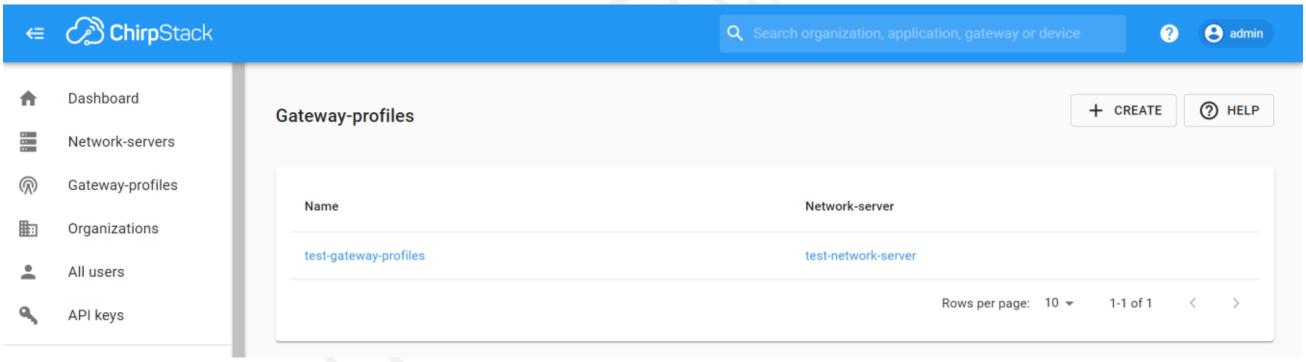


Name *
test-gateway-profiles
A short name identifying the gateway-profile.

Enabled channels *
0, 1, 2
The channels active in this gateway-profile as specified in the LoRaWAN Regional Parameters specification. Separate channels by comma, e.g. 0, 1, 2. Extra channels must not be included in this list.

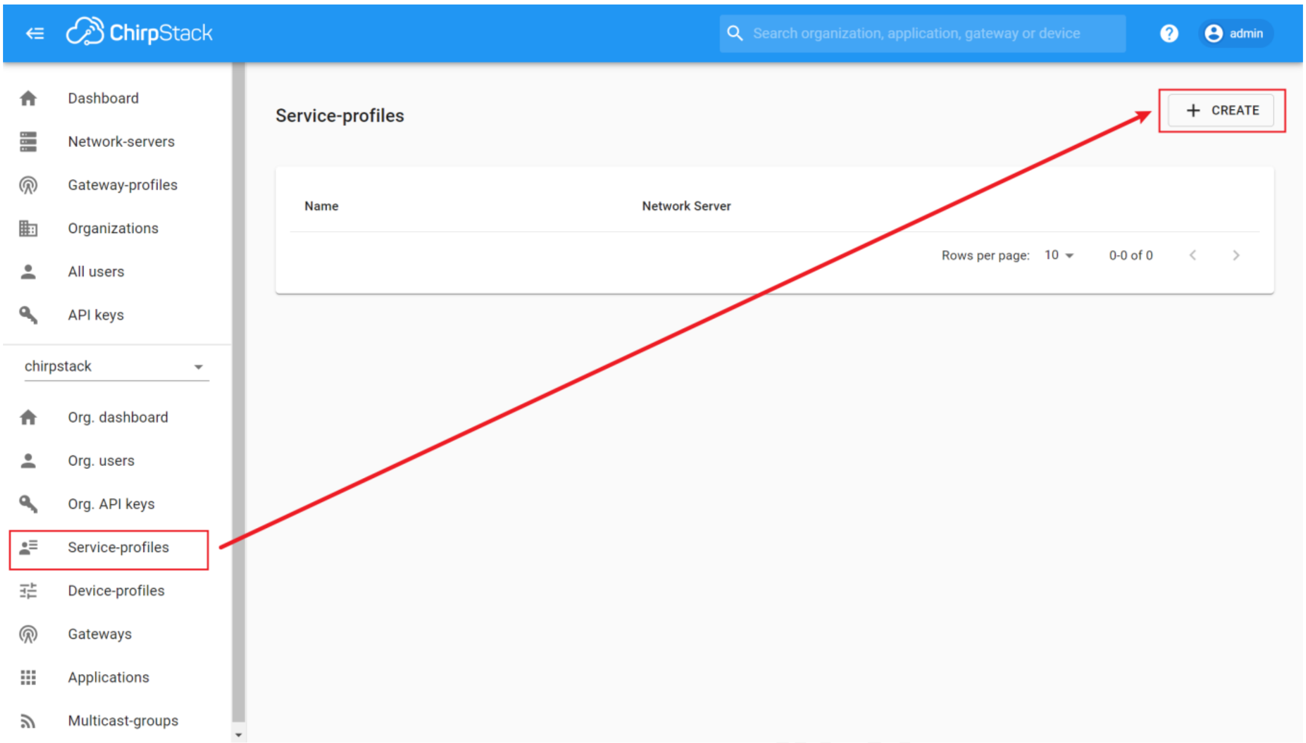
Network-server *
test-network-server

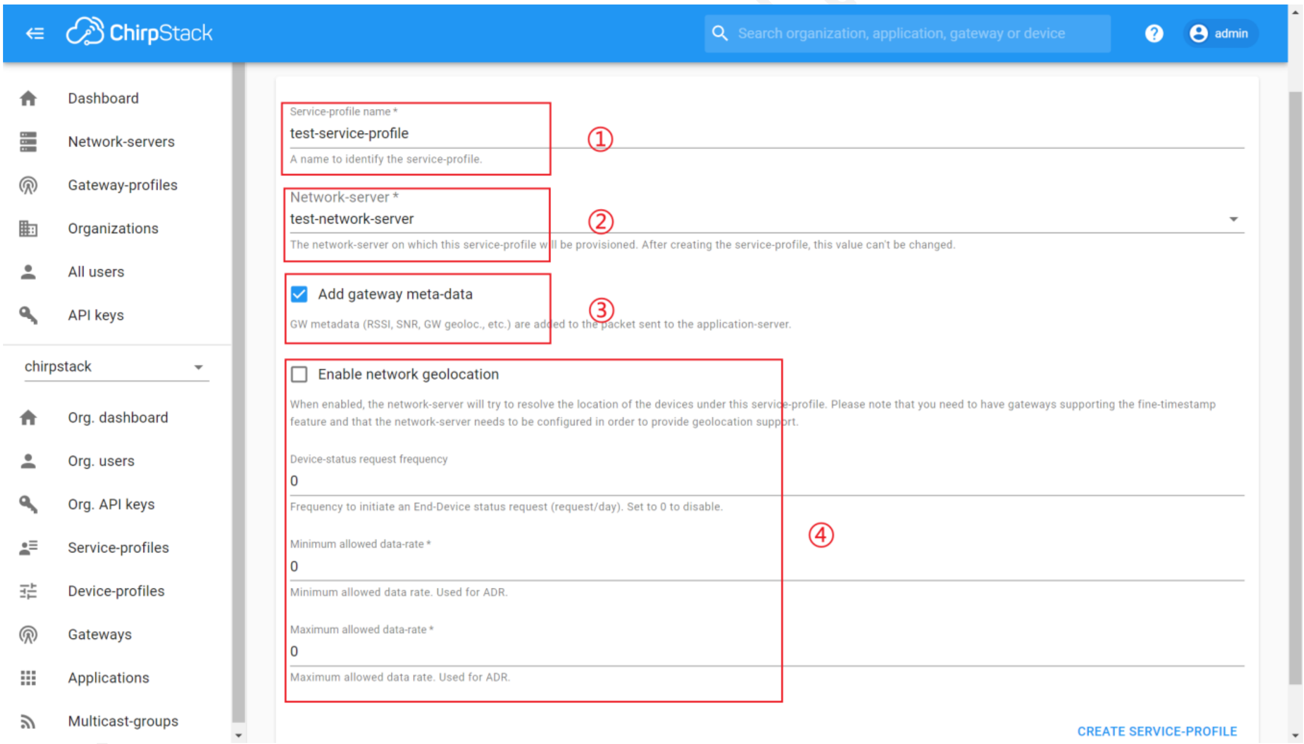
Click the "GREATE GATEWAY-PROFILE".



Name	Network-server
test-gateway-profiles	test-network-server

5.4.4 Create the Service-profiles





Service-profile name *
test-service-profile
A name to identify the service-profile.

Network-server *
test-network-server
The network-server on which this service-profile will be provisioned. After creating the service-profile, this value can't be changed.

☑ Add gateway meta-data
GW metadata (RSSI, SNR, GW geoloc., etc.) are added to the packet sent to the application-server.

☐ Enable network geolocation
When enabled, the network-server will try to resolve the location of the devices under this service-profile. Please note that you need to have gateways supporting the fine-timestamp feature and that the network-server needs to be configured in order to provide geolocation support.

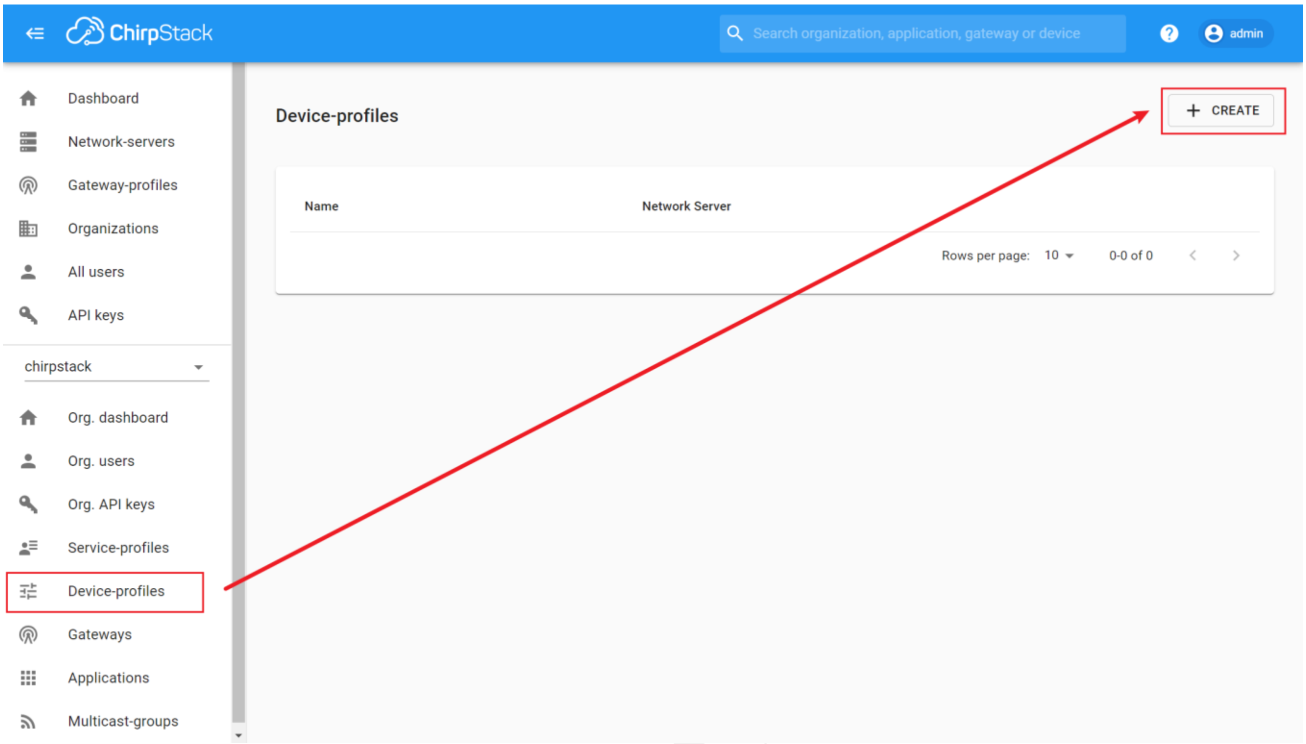
Device-status request frequency
0
Frequency to initiate an End-Device status request (request/day). Set to 0 to disable.

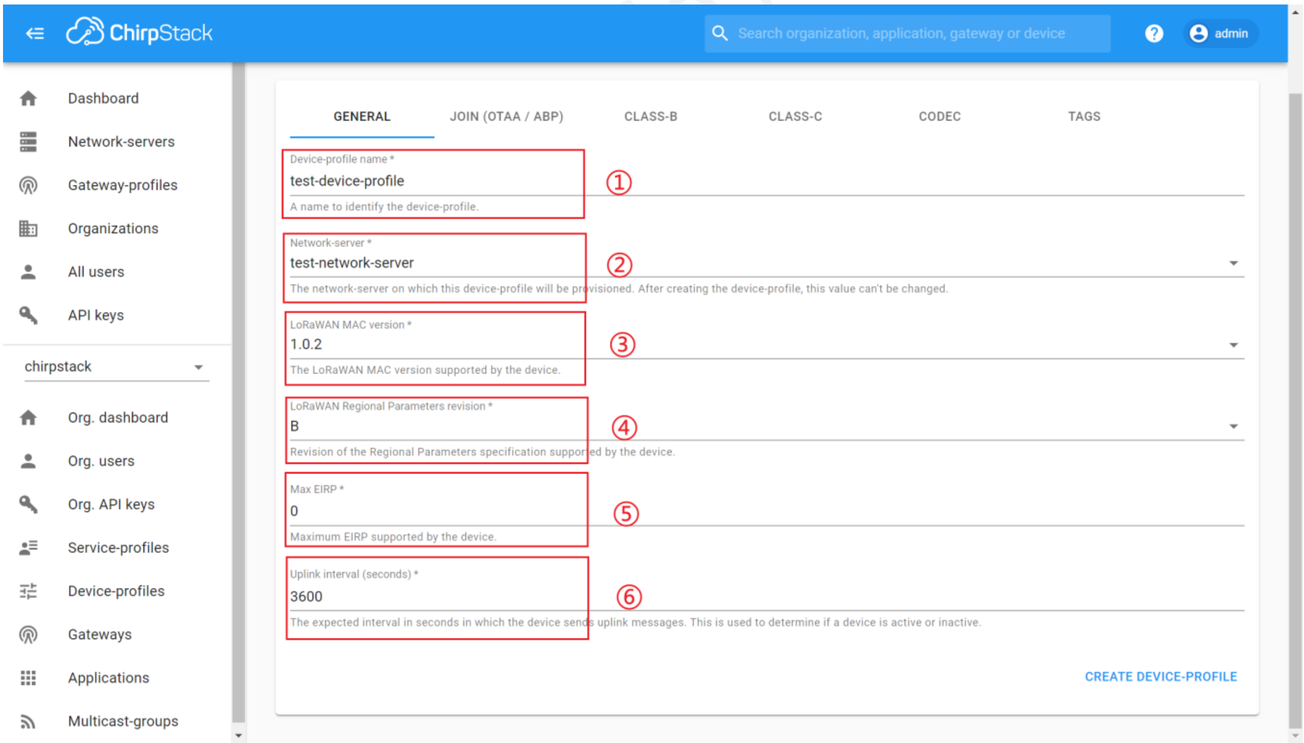
Minimum allowed data-rate *
0
Minimum allowed data rate. Used for ADR.

Maximum allowed data-rate *
0
Maximum allowed data rate. Used for ADR.

① Service-profile name: custom name.
② Network-server: select the Network-server you created earlier.
③ Add gateway meta-data: select it.
④ Default values are usually used.

5.4.5 Create the Device-profiles





Device-profile name * test-device-profile ①
A name to identify the device-profile.

Network-server * test-network-server ②
The network-server on which this device-profile will be provisioned. After creating the device-profile, this value can't be changed.

LoRaWAN MAC version * 1.0.2 ③
The LoRaWAN MAC version supported by the device.

LoRaWAN Regional Parameters revision * B ④
Revision of the Regional Parameters specification supported by the device.

Max EIRP * 0 ⑤
Maximum EIRP supported by the device.

Uplink interval (seconds) * 3600 ⑥
The expected interval in seconds in which the device sends uplink messages. This is used to determine if a device is active or inactive.

① Device-profile name: custom name.

② Network-server: select the Network-server you created earlier.

③ LoRaWAN MAC version: 1.0.2 (only for SenseCAP Node)

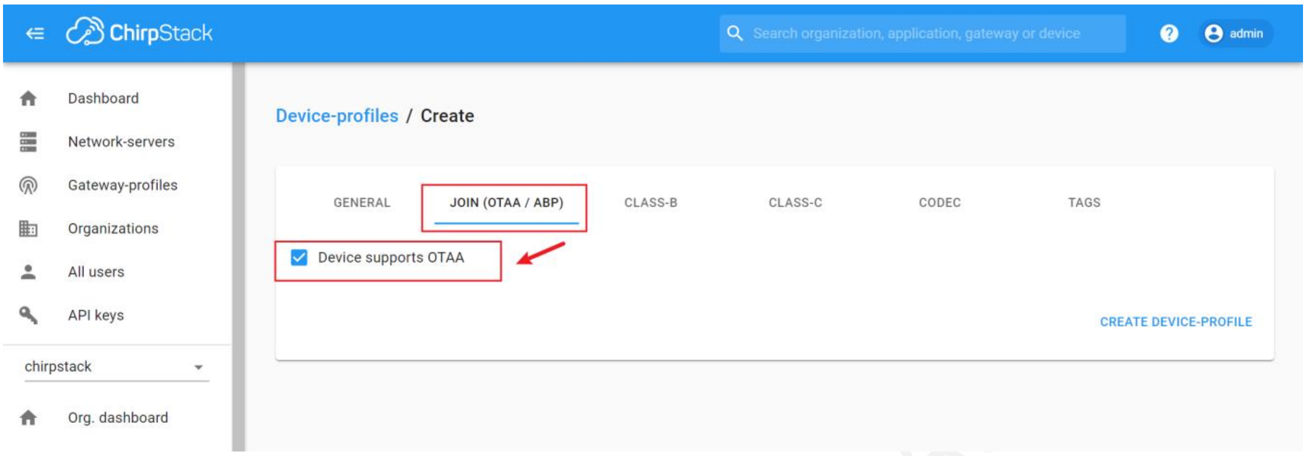
④ LoRaWAN Regional Parameters revision: B (only for SenseCAP Node)

⑤ Max EIRP: 0

⑥ Uplink interval (seconds) : 3600

Be consistent with the node's upload interval.

Click the “JOIN(OTAA/ABP)”, and select “Device supports OTAA”.



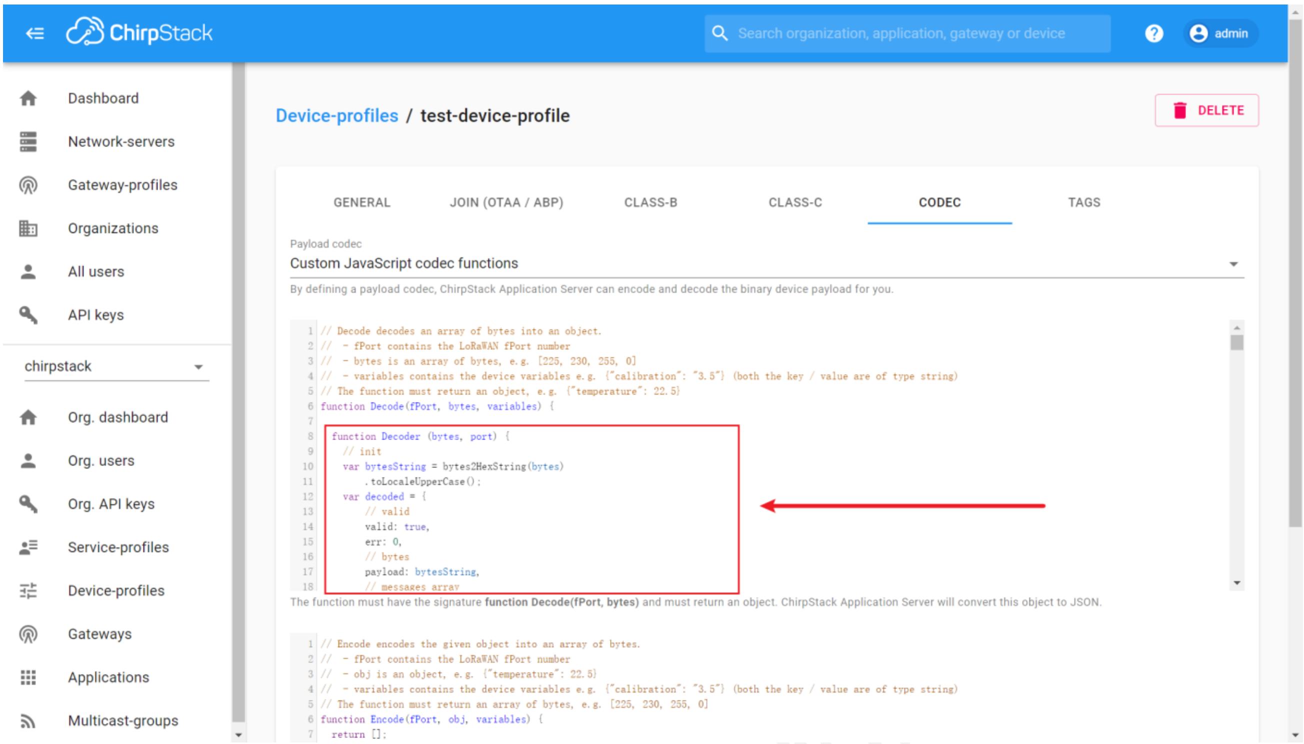
☑ Device supports OTAA

To get a SenseCAP Sensor Node on quick decoding, we provide a piece of code.

Click the “CODEC”, and select “Custom JavaScript codec functions”.

Then view https://github.com/Seeed-Solution/TTN-Payload-Decoder/blob/master/decoder.js , please copy the code to “function decode” FUNC.

```
function Decoder (bytes, port) {
   // init
   var bytesString = bytes2HexString(bytes)
         .toLocaleUpperCase();
……….

return binaryData.toString()
            .replace(/,/g, "");
}
```



Payload codec
Custom JavaScript codec functions

By defining a payload codec, ChirpStack Application Server can encode and decode the binary device payload for you.

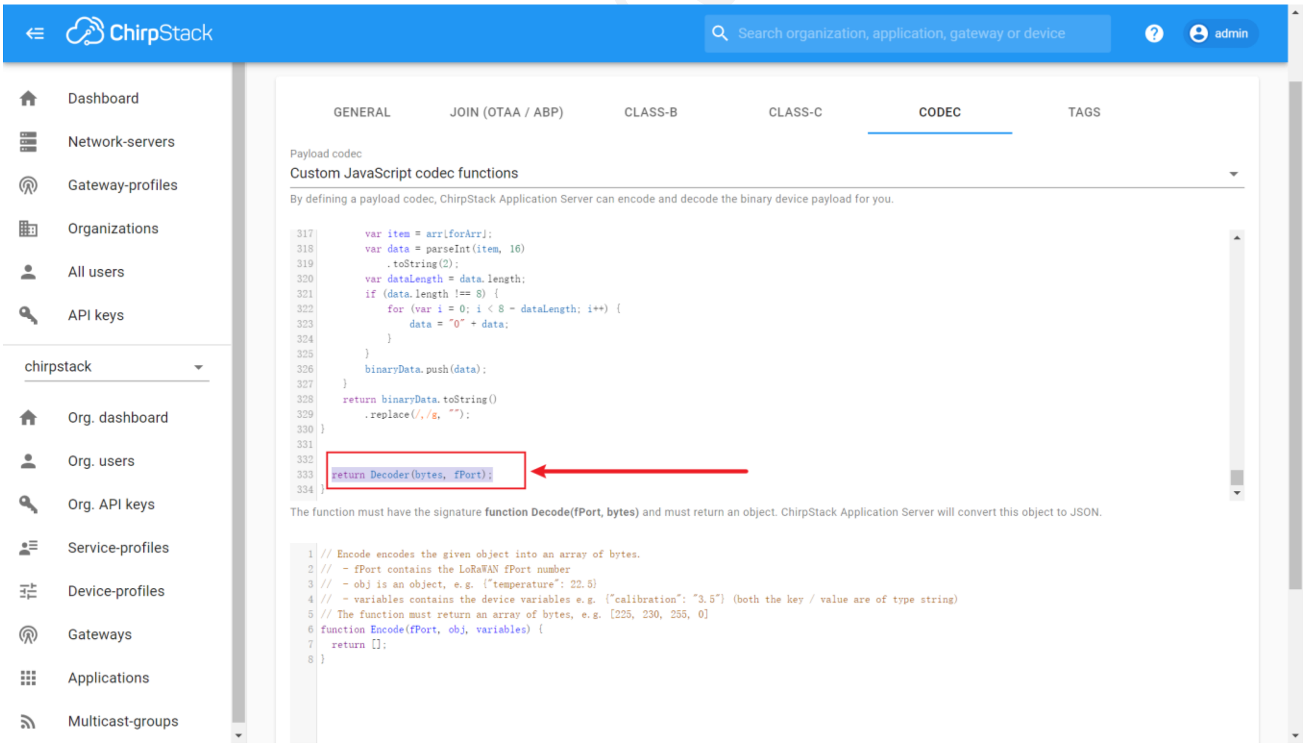
```
// Decode decodes an array of bytes into an object.
//  - fPort contains the LoRaWAN fPort number
//  - bytes is an array of bytes, e.g. [225, 230, 255, 0]
//  - variables contains the device variables e.g. {"calibration": "3.5"} (both the key / value are of type string)
// The function must return an object, e.g. {"temperature": 22.5}
function Decode(fPort, bytes, variables) {

 function Decoder (bytes, port) {
   // init
   var bytesString = bytes2HexString(bytes)
       .toLocaleUpperCase();
   var decoded = {
       // valid
       valid: true,
       err: 0,
       // bytes
       payload: bytesString,
       // messages array
```

The function must have the signature **function Decode(fPort, bytes)** and must return an object. ChirpStack Application Server will convert this object to JSON.

```
// Encode encodes the given object into an array of bytes.
//  - fPort contains the LoRaWAN fPort number
//  - obj is an object, e.g. {"temperature": 22.5}
//  - variables contains the device variables e.g. {"calibration": "3.5"} (both the key / value are of type string)
// The function must return an array of bytes, e.g. [225, 230, 255, 0]
function Encode(fPort, obj, variables) {
  return [];
```

Add the return value at the end:

```
return Decoder(bytes, fPort);
```



Payload codec
Custom JavaScript codec functions

By defining a payload codec, ChirpStack Application Server can encode and decode the binary device payload for you.

```
        var item = arr[forArr];
        var data = parseInt(item, 16)
            .toString(2);
        var dataLength = data.length;
        if (data.length !== 8) {
            for (var i = 0; i < 8 - dataLength; i++) {
                data = "0" + data;
            }
        }
        binaryData.push(data);
    }
    return binaryData.toString()
        .replace(/,/g, "");
}


 return Decoder(bytes, fPort);
}
```

The function must have the signature **function Decode(fPort, bytes)** and must return an object. ChirpStack Application Server will convert this object to JSON.

```
// Encode encodes the given object into an array of bytes.
//  - fPort contains the LoRaWAN fPort number
//  - obj is an object, e.g. {"temperature": 22.5}
//  - variables contains the device variables e.g. {"calibration": "3.5"} (both the key / value are of type string)
// The function must return an array of bytes, e.g. [225, 230, 255, 0]
function Encode(fPort, obj, variables) {
  return [];
}
```

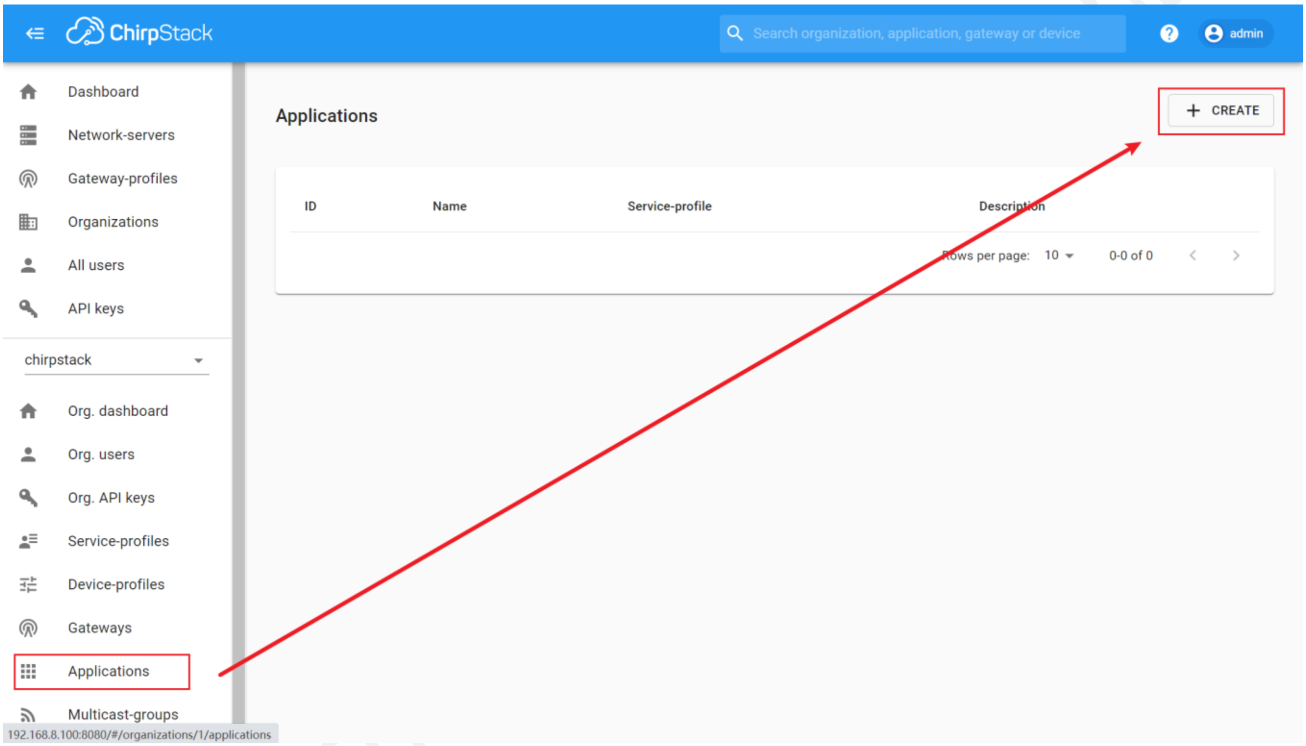
Finally, click "Create".

5.5 Add Sensor Node to ChirpStack LoRa Server

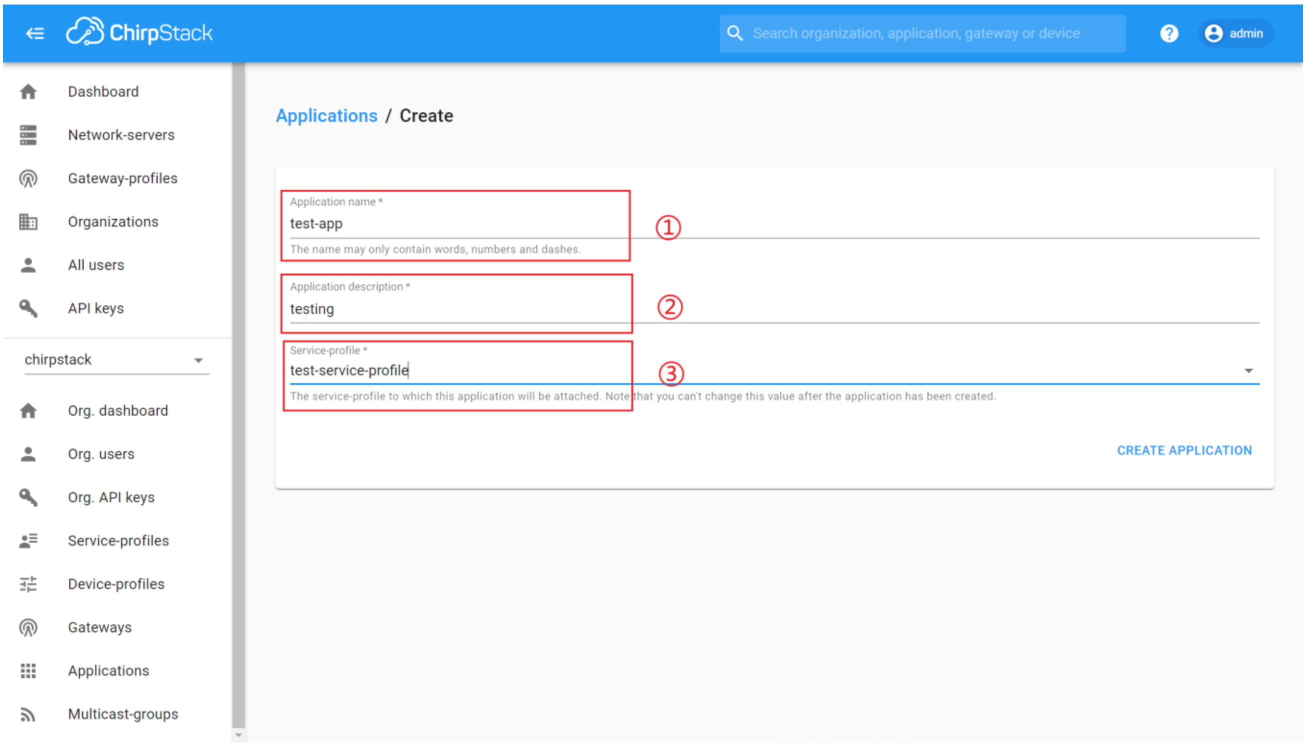
5.5.1 Get Node's EUI and Key

Refer to section 3.1.

5.5.2 Create an Application



① Application name: custom name.
② Application description: custom description.
③ Service-profile: select the Service-profile you created earlier.



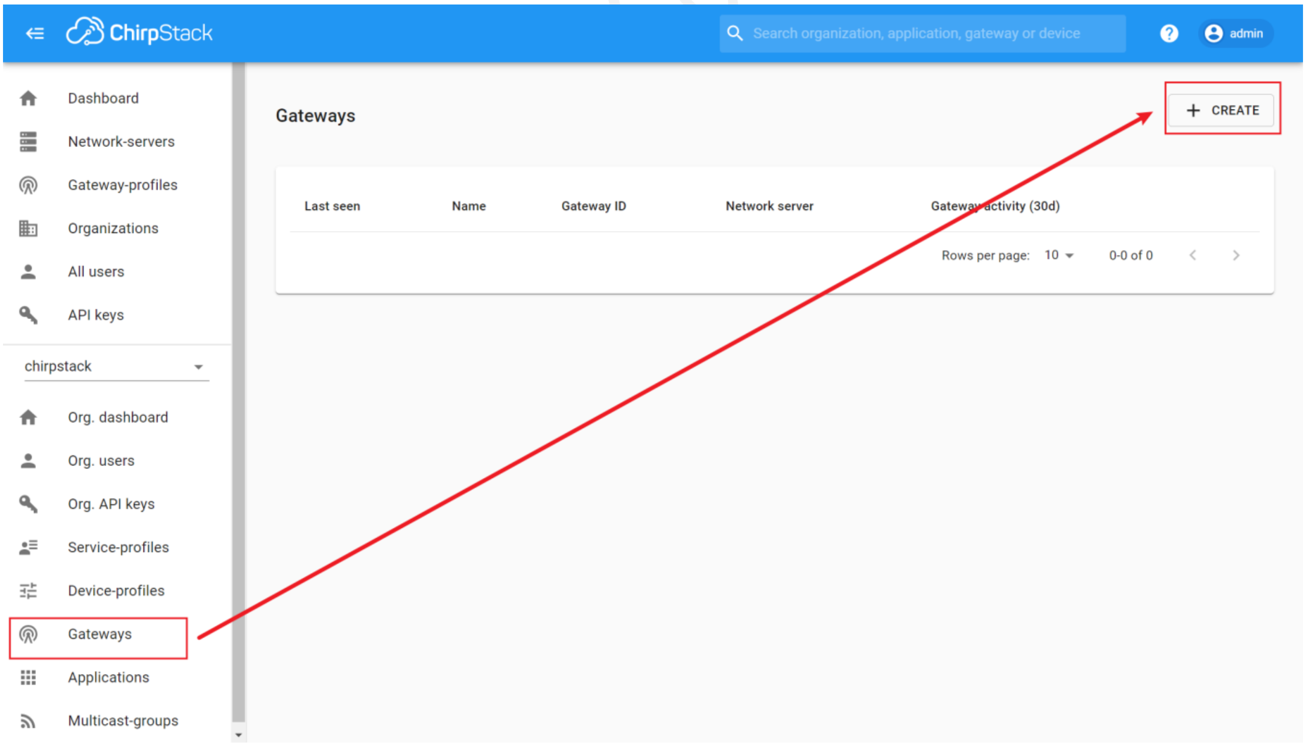
Applications / Create

Application name *
test-app ①
The name may only contain words, numbers and dashes.

Application description *
testing ②

Service-profile *
test-service-profile ③
The service-profile to which this application will be attached. Note that you can't change this value after the application has been created.

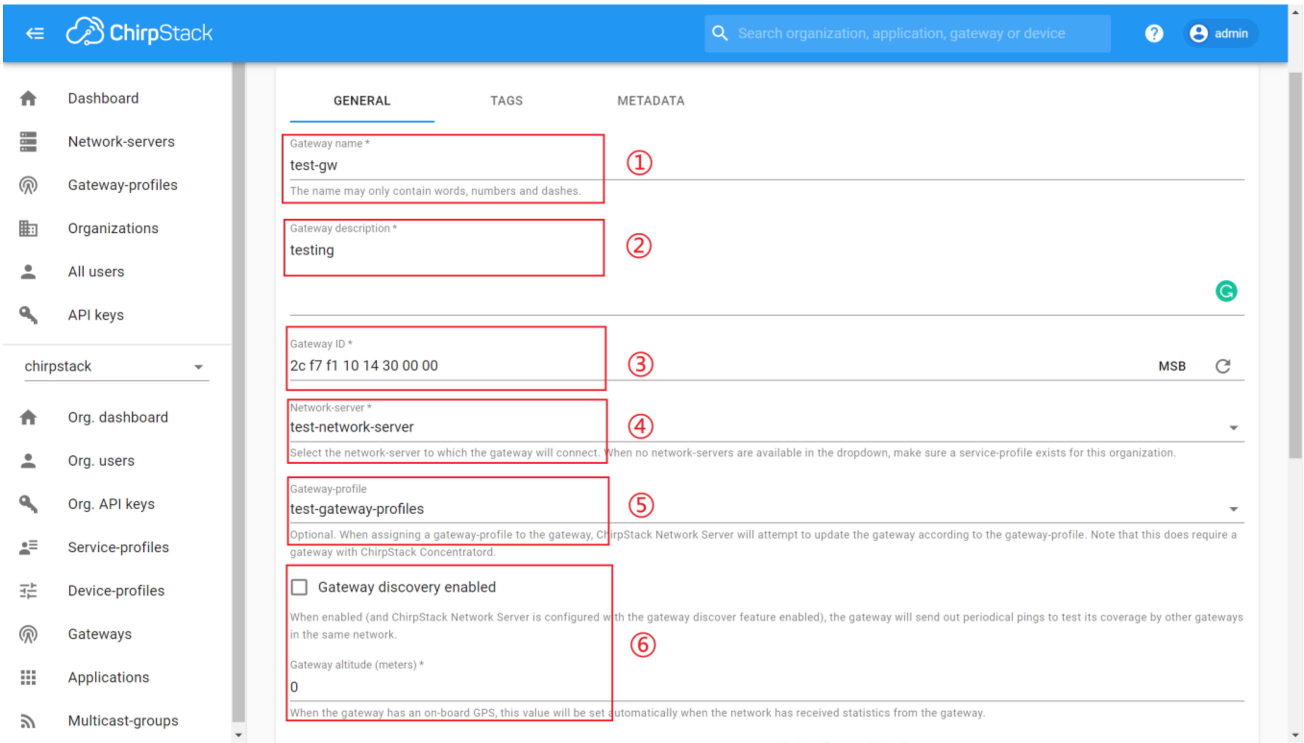
5.5.3 Create a Gateway



Gateways

Last seen	Name	Gateway ID	Network server	Gateway activity (30d)

Rows per page: 10 0-0 of 0



Gateway name *
test-gw
The name may only contain words, numbers and dashes.

Gateway description *
testing

Gateway ID *
2c f7 f1 10 14 30 00 00

Network-server *
test-network-server
Select the network-server to which the gateway will connect. When no network-servers are available in the dropdown, make sure a service-profile exists for this organization.

Gateway-profile
test-gateway-profiles
Optional. When assigning a gateway-profile to the gateway, ChirpStack Network Server will attempt to update the gateway according to the gateway-profile. Note that this does require a gateway with ChirpStack Concentratord.

☐ Gateway discovery enabled
When enabled (and ChirpStack Network Server is configured with the gateway discover feature enabled), the gateway will send out periodical pings to test its coverage by other gateways in the same network.

Gateway altitude (meters) *
0
When the gateway has an on-board GPS, this value will be set automatically when the network has received statistics from the gateway.

① Gateway name: custom name.

② Gateway description: custom description.

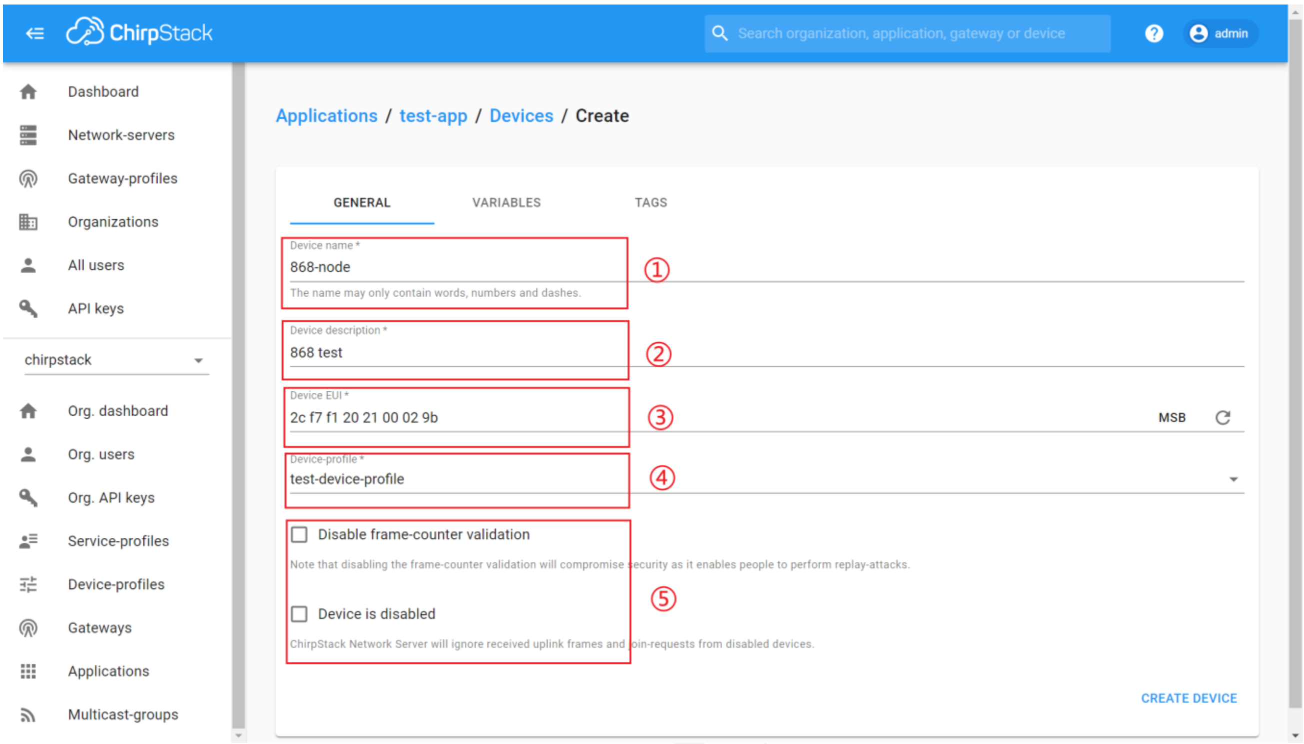
③ Gateway ID: the gateway EUI, see the gateway’s label.

④ Network-server: select the Network-server you created earlier.

⑤ Gateway-profile: select the Gateway-profile you created earlier.

⑥ Default values are usually used.

5.5.4 Create a Device



Applications / test-app / Devices / Create

Device name *
868-node
The name may only contain words, numbers and dashes.

Device description *
868 test

Device EUI *
2c f7 f1 20 21 00 02 9b

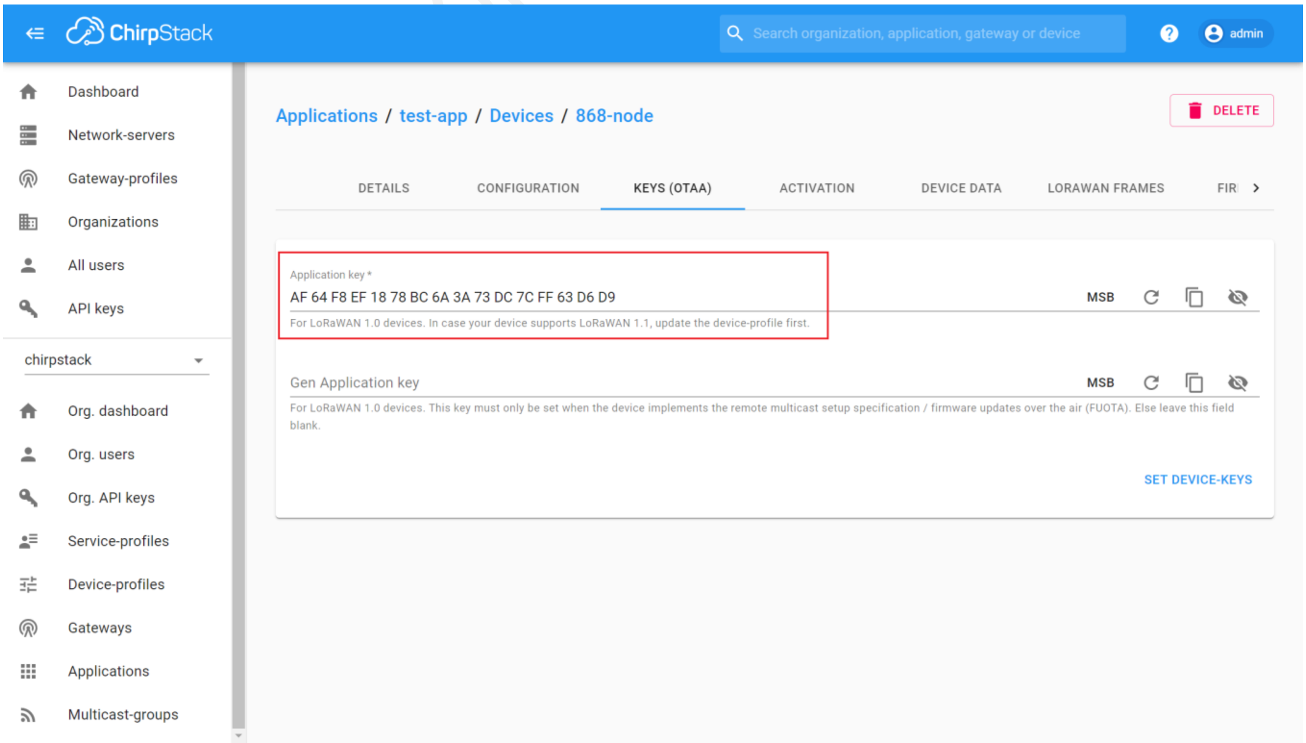
Device-profile *
test-device-profile

□ Disable frame-counter validation
Note that disabling the frame-counter validation will compromise security as it enables people to perform replay-attacks.

□ Device is disabled
ChirpStack Network Server will ignore received uplink frames and join-requests from disabled devices.

① Device name: custom name.
② Device description: custom description.
③ Device EUI: the Node’s EUI.
④ Device-profile: select the Device-profile you created earlier.
⑤ Don't check and ignore it.

Click “Create” and enter the App KEY (Application Key, refer to section 3.1).



Applications / test-app / Devices / 868-node

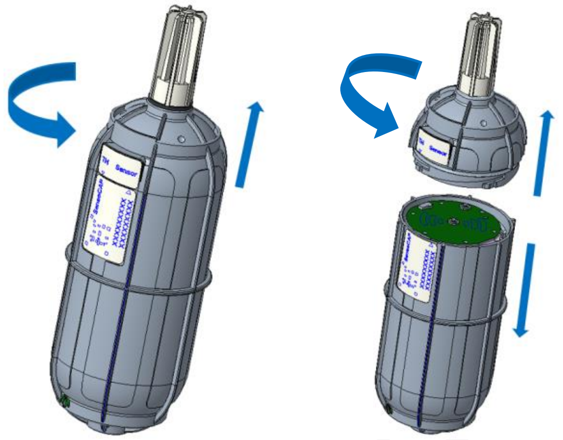
Application key *
AF 64 F8 EF 18 78 BC 6A 3A 73 DC 7C FF 63 D6 D9
For LoRaWAN 1.0 devices. In case your device supports LoRaWAN 1.1, update the device-profile first.

Gen Application key
For LoRaWAN 1.0 devices. This key must only be set when the device implements the remote multicast setup specification / firmware updates over the air (FUOTA). Else leave this field blank.

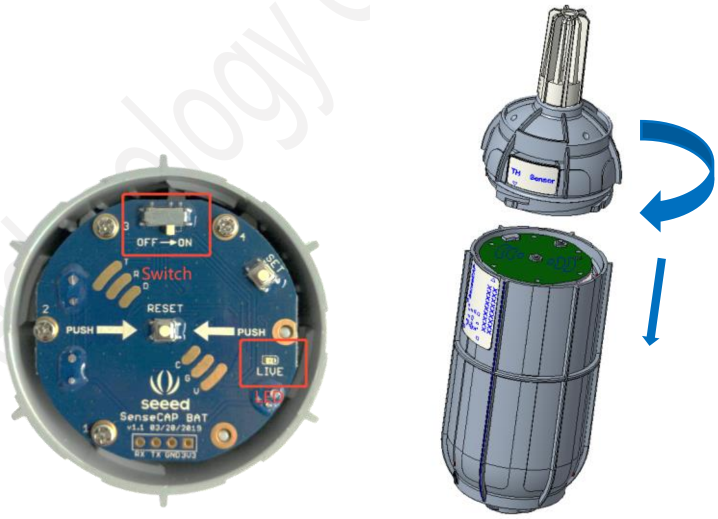
5.5.5 Power on

The power switch is hidden inside the device. Open the device and turn on the power before installing the sensors. Here is the step-by-step instruction:

4) Loosen the Sensor Probe by turning the cap counterclockwise. Use the white cap opener to make this process easier. The image below uses TH Sensor as an example and applies to all other SenseCAP sensors.



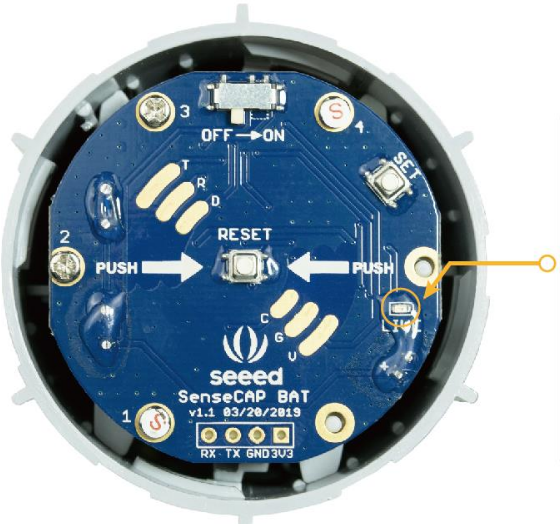
5) After opening the device, turn the switch to “ON”, and the LED on the lower right corner will flash, indicating that the power is on. Wait for about 10 seconds, then the LED will flash quickly for 2 seconds, indicating that the device is connected to the network.



6) After the device is connected to the network, connect the Sensor Probe back with the Sensor Node Controller by turning it clockwise. Please note that the labels on both parts should be aligned as shown in the image below, otherwise the two parts will not be attached to function properly and data will not be uploaded.

5.5.6 Sensor Node Working Status

You can refer to the LED indicator for the Sensor Node for its working status. Please see the status explanations in the image below:



LED Status

After powering on the device

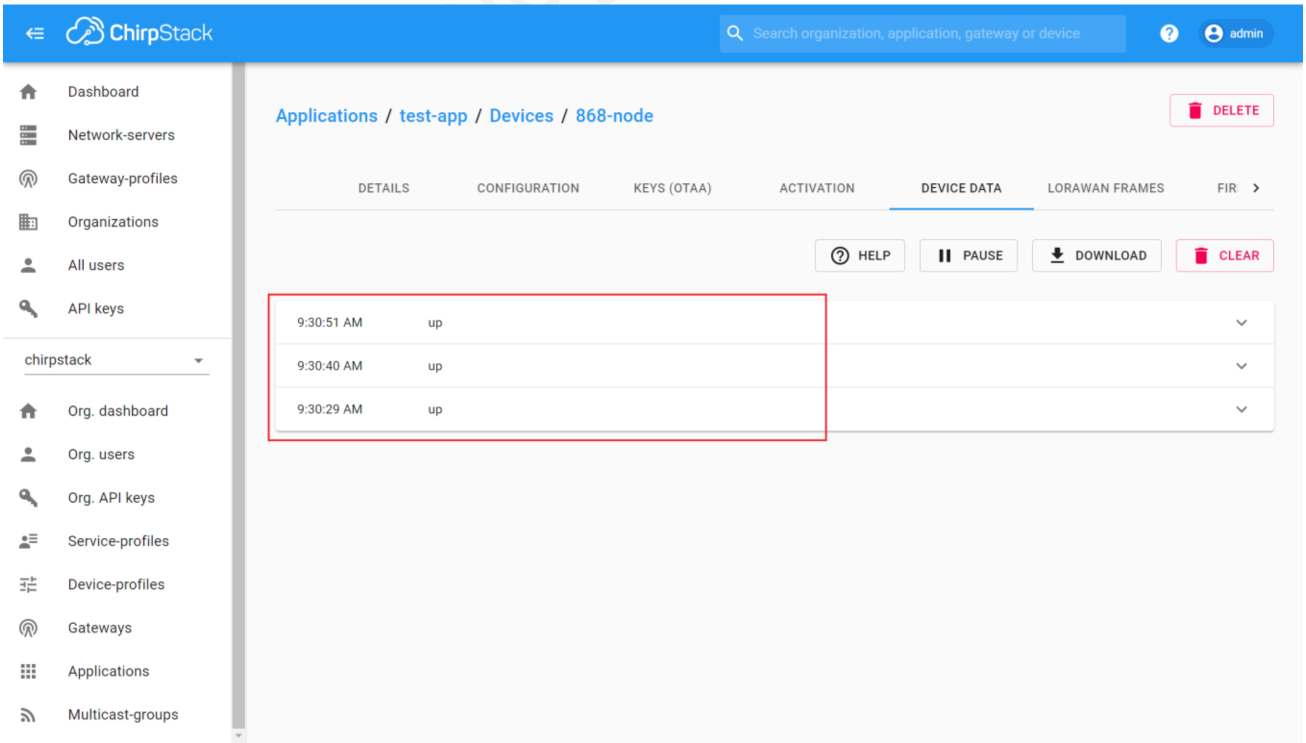
1. LED flashes once after powering on, then turn OFF
2. After 10 seconds, LED flashes quickly for 2 seconds, indicating it has joined the network
3. After joining the network, the LED stays off to save energy
4. Push the reset button to re-join the network if the LED does not start flashing 15 seconds after powering on

5.5.7 Checking Data Upload

On the "DEVICE DATA" page in the web, you can view the data that the gateway has received from the Sensor Node.

To get measurement ID information, please visit :

https://sensecap-docs.seeed.cc/sensor_types_list.html



Applications / test-app / Devices / 868-node

9:30:51 AM up

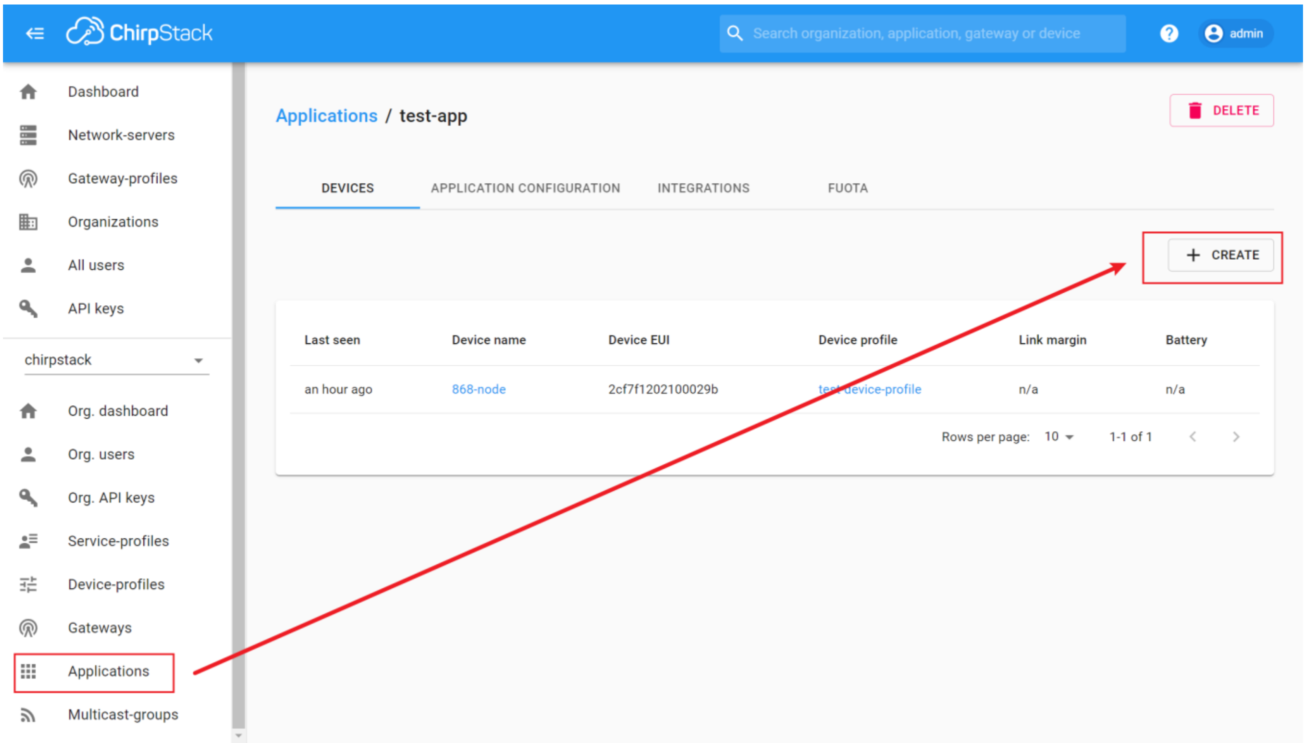
9:30:40 AM up

9:30:29 AM up

5.6 Add a 3rd Part Node Device

(1) Refer to the previous section to configure the gateway.

(2) Add a new device to Application.

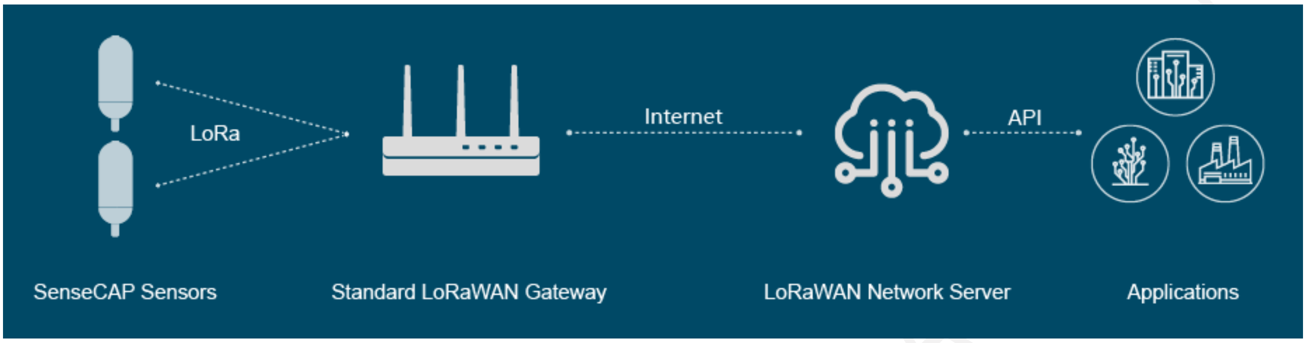


Last seen	Device name	Device EUI	Device profile	Link margin	Battery
an hour ago	868-node	2cf7f1202100029b	test-device-profile	n/a	n/a

(3) Refer to data parsing and tutorials for third-party devices.

6 The Node Connects to a Standard Gateway

SenseCAP Sensor Nodes are designed on The Things Network LoRaWAN servers, the firmware supports standard LoRaWAN 1.0.2 protocol, making it possible to connect to other 3rd-party LoRaWAN gateways and servers.



6.1 Node Frequency Plans

Frequency Plans	
EU868	Uplink: 868.1 - SF7BW125 to SF12BW125 868.3 - SF7BW125 to SF12BW125 and SF7BW250 868.5 - SF7BW125 to SF12BW125 867.1 - SF7BW125 to SF12BW125 867.3 - SF7BW125 to SF12BW125 867.5 - SF7BW125 to SF12BW125 867.7 - SF7BW125 to SF12BW125 867.9 - SF7BW125 to SF12BW125 868.8 – FSK Downlink: Uplink channels 1-9 (RX1) 869.525 - SF9BW125 (RX2 downlink only)
US915	Uplink: 903.9 - SF7BW125 to SF10BW125 904.1 - SF7BW125 to SF10BW125 904.3 - SF7BW125 to SF10BW125 904.5 - SF7BW125 to SF10BW125 904.7 - SF7BW125 to SF10BW125 904.9 - SF7BW125 to SF10BW125 905.1 - SF7BW125 to SF10BW125

	905.3 - SF7BW125 to SF10BW125 904.6 - SF8BW500 Downlink: 923.3 - SF7BW500 to SF12BW500 923.9 - SF7BW500 to SF12BW500 924.5 - SF7BW500 to SF12BW500 925.1 - SF7BW500 to SF12BW500 925.7 - SF7BW500 to SF12BW500 926.3 - SF7BW500 to SF12BW500 926.9 - SF7BW500 to SF12BW500 927.5 - SF7BW500 to SF12BW500

6.2 A Standard LoRaWAN Gateway Configuration Example

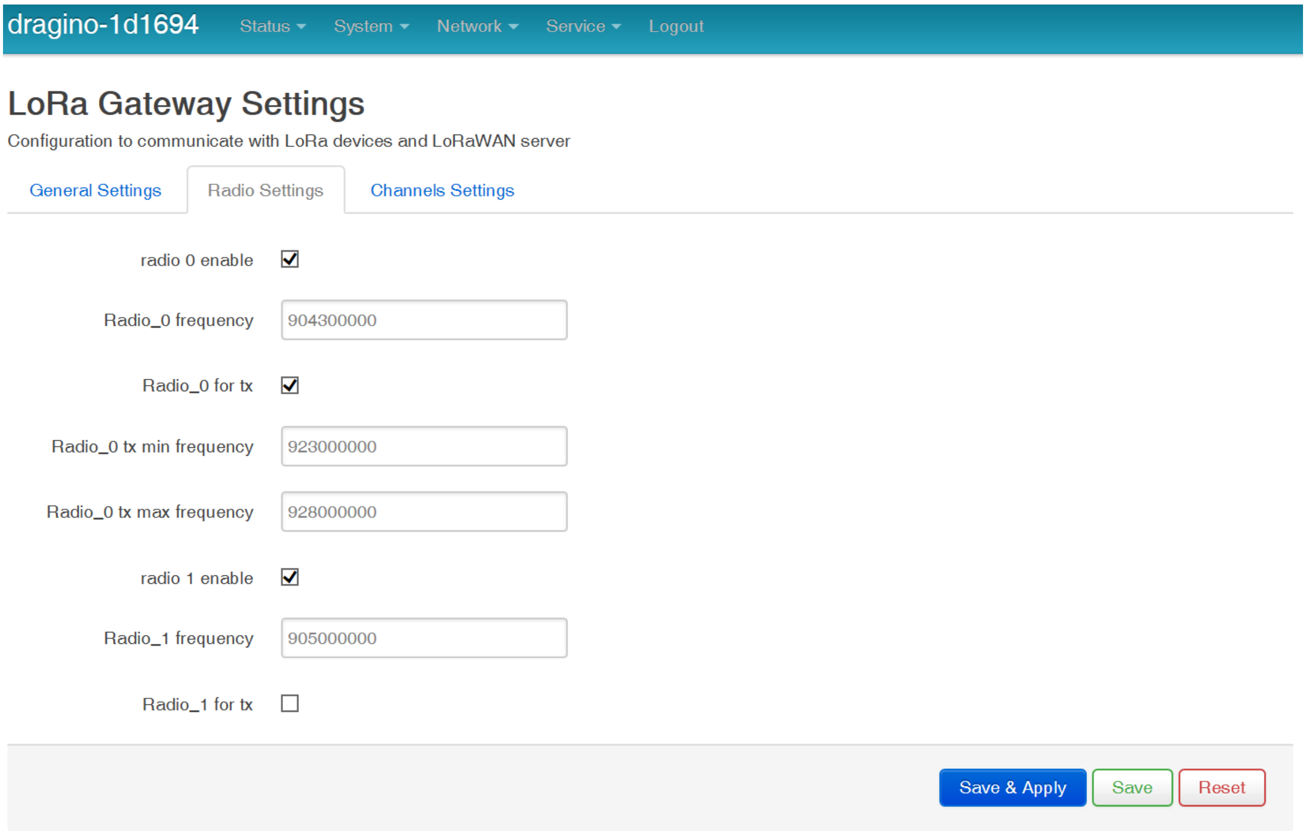
Typically, the LoRaWAN gateway needs to set the server address and uplink and downlink channel parameters for the end device. Refer to the gateway user manual to configure the server. Here, a common LoRaWAN gateway (US915) is taken as an example to explain how to configure the communication parameters of the Sensor Node.

The detailed configuration parameters for the Sensor Node are described here:
https://github.com/Jenkinlu001/SenseCAP-LoRaWAN/tree/master/LoRaWAN_Node_Parameters

6.2.1 Radio Settings

Find radio settings or frequency settings in the background of the gateway.

```
radio 0 enable√

Radio_0 frequency: 904300000

Radio_0 for tx√

Radio_0 tx min frequency: 923000000
Radio_0 tx max frequency: 928000000

radio 1 enable√

Radio_1 frequency: 905000000
```



LoRa Gateway Settings

Configuration to communicate with LoRa devices and LoRaWAN server

radio 0 enable ☑

Radio_0 frequency 904300000

Radio_0 for tx ☑

Radio_0 tx min frequency 923000000

Radio_0 tx max frequency 928000000

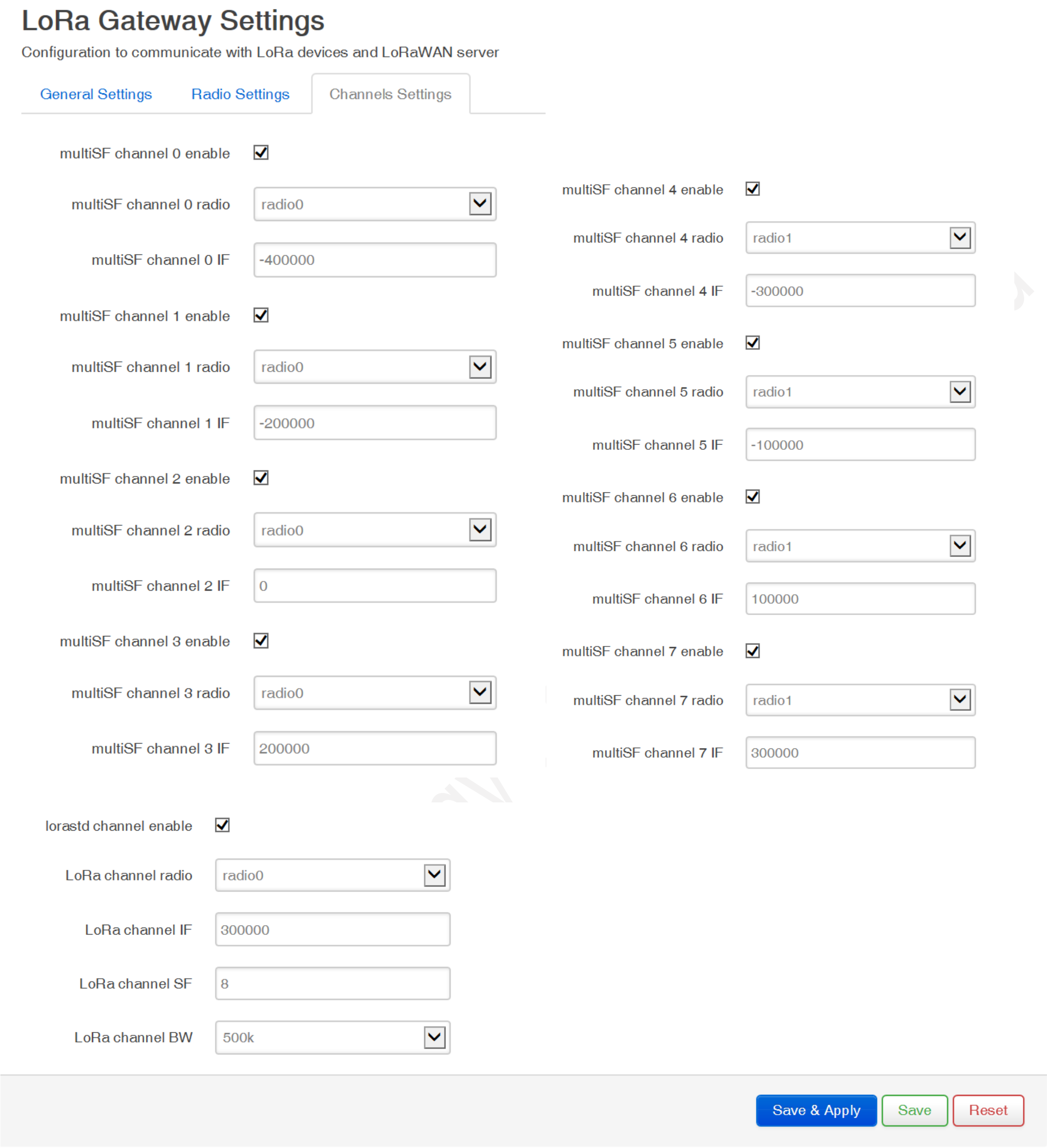
radio 1 enable ☑

Radio_1 frequency 905000000

Radio_1 for tx □

6.2.2 Channel Settings

Please refer to the items in the following image for channel settings.



LoRa Gateway Settings

Configuration to communicate with LoRa devices and LoRaWAN server

multiSF channel 0 enable ☑

multiSF channel 0 radio: radio0

multiSF channel 0 IF: -400000

multiSF channel 1 enable ☑

multiSF channel 1 radio: radio0

multiSF channel 1 IF: -200000

multiSF channel 2 enable ☑

multiSF channel 2 radio: radio0

multiSF channel 2 IF: 0

multiSF channel 3 enable ☑

multiSF channel 3 radio: radio0

multiSF channel 3 IF: 200000

multiSF channel 4 enable ☑

multiSF channel 4 radio: radio1

multiSF channel 4 IF: -300000

multiSF channel 5 enable ☑

multiSF channel 5 radio: radio1

multiSF channel 5 IF: -100000

multiSF channel 6 enable ☑

multiSF channel 6 radio: radio1

multiSF channel 6 IF: 100000

multiSF channel 7 enable ☑

multiSF channel 7 radio: radio1

multiSF channel 7 IF: 300000

lorastd channel enable ☑

LoRa channel radio: radio0

LoRa channel IF: 300000

LoRa channel SF: 8

LoRa channel BW: 500k

6.2.3 Power on

Refer to section 4.5.5

6.2.4 Sensor Node Working Status

Refer to section 4.5.6

6.2.5 Checking Data Upload

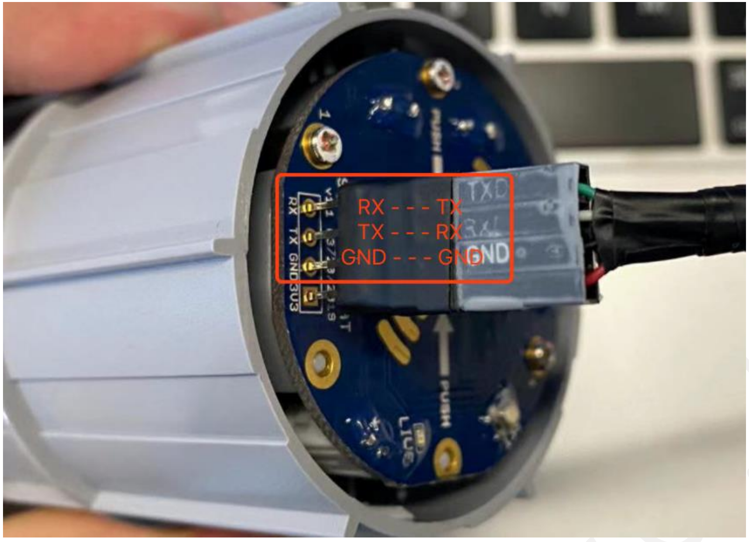
On the log page in the background of the gateway, you can view the data that the gateway has received from the Sensor Node.

6.3 Modify Node's EUI, KEY, and Duty

Connect serial ports (as shown in the image below), turn on the power, launch the serial port monitoring tool on your computer, set the Baud Rate as 115200.

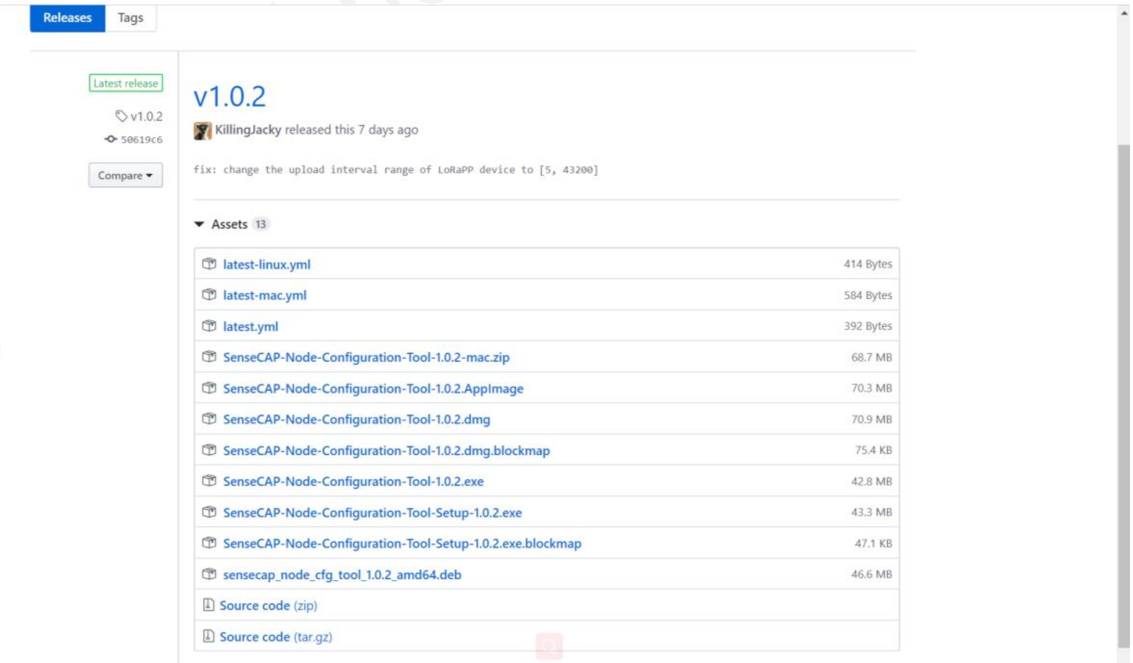
(1) Use the USB to TTL wire (Please leave power port, aka 3V3 unconnected):

```
TX---RX
RX---TX
GND---GND
```



(2) Install the Serial Tool. Download via: https://github.com/Seeed-Solution/SenseCAP-Node-Configuration-Tool/releases/tag/v1.0.2

Windows: SenseCAP-Node-Configuration-Tool-1.x.x.exe
Mac: SenseCAP-Node-Configuration-Tool-1.0.2-mac.zip



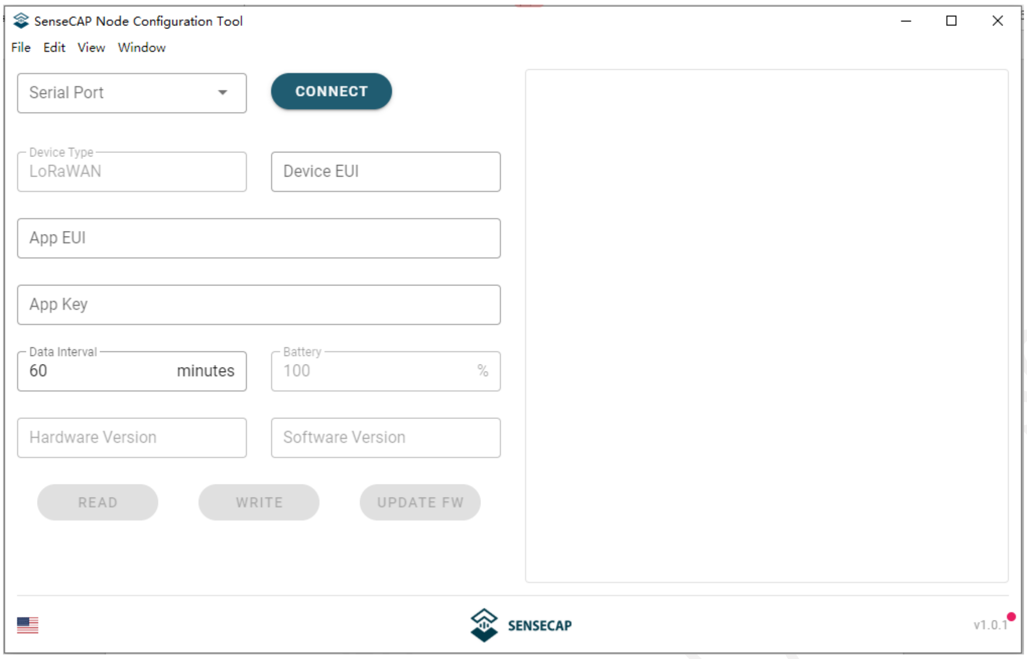
v1.0.2

KillingJacky released this 7 days ago

```
fix: change the upload interval range of LoRaPP device to [5, 43200]
```

▾ Assets 13

latest-linux.yml	414 Bytes
latest-mac.yml	584 Bytes
latest.yml	392 Bytes
SenseCAP-Node-Configuration-Tool-1.0.2-mac.zip	68.7 MB
SenseCAP-Node-Configuration-Tool-1.0.2.AppImage	70.3 MB
SenseCAP-Node-Configuration-Tool-1.0.2.dmg	70.9 MB
SenseCAP-Node-Configuration-Tool-1.0.2.dmg.blockmap	75.4 KB
SenseCAP-Node-Configuration-Tool-1.0.2.exe	42.8 MB
SenseCAP-Node-Configuration-Tool-Setup-1.0.2.exe	43.3 MB
SenseCAP-Node-Configuration-Tool-Setup-1.0.2.exe.blockmap	47.1 KB
sensecap_node_cfg_tool_1.0.2_amd64.deb	46.6 MB
Source code (zip)	
Source code (tar.gz)	



Serial Port

Device Type: LoRaWAN

Device EUI

App EUI

App Key

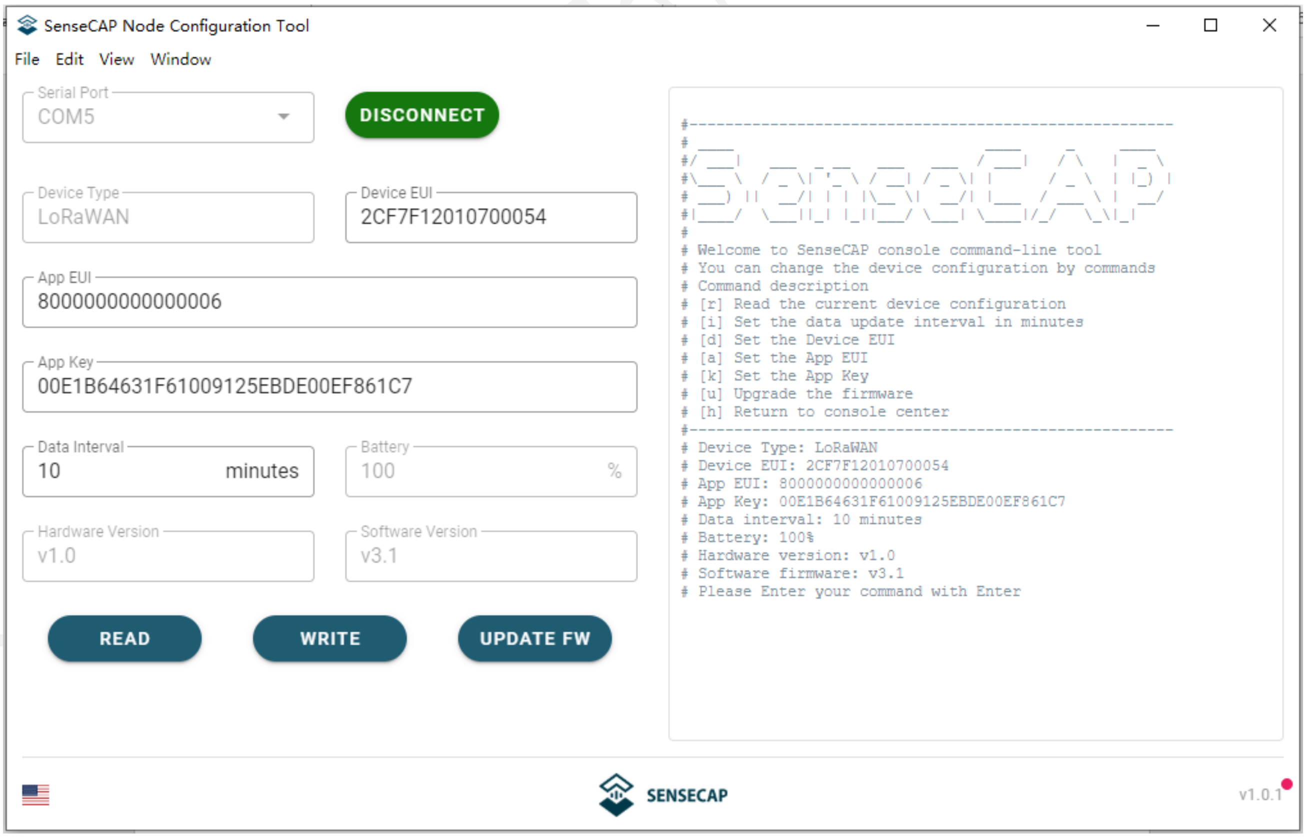
Data Interval: 60 minutes

Battery: 100 %

Hardware Version

Software Version

(3) Select the COM Port that your tool uses, and click "CONNECT".
Press "SET" button on the Sensor Controller, meanwhile flip the switch to "ON", and you will see "SenseCAP".



Serial Port: COM5

Device Type: LoRaWAN

Device EUI: 2CF7F12010700054

App EUI: 8000000000000006

App Key: 00E1B64631F61009125EBDE00EF861C7

Data Interval: 10 minutes

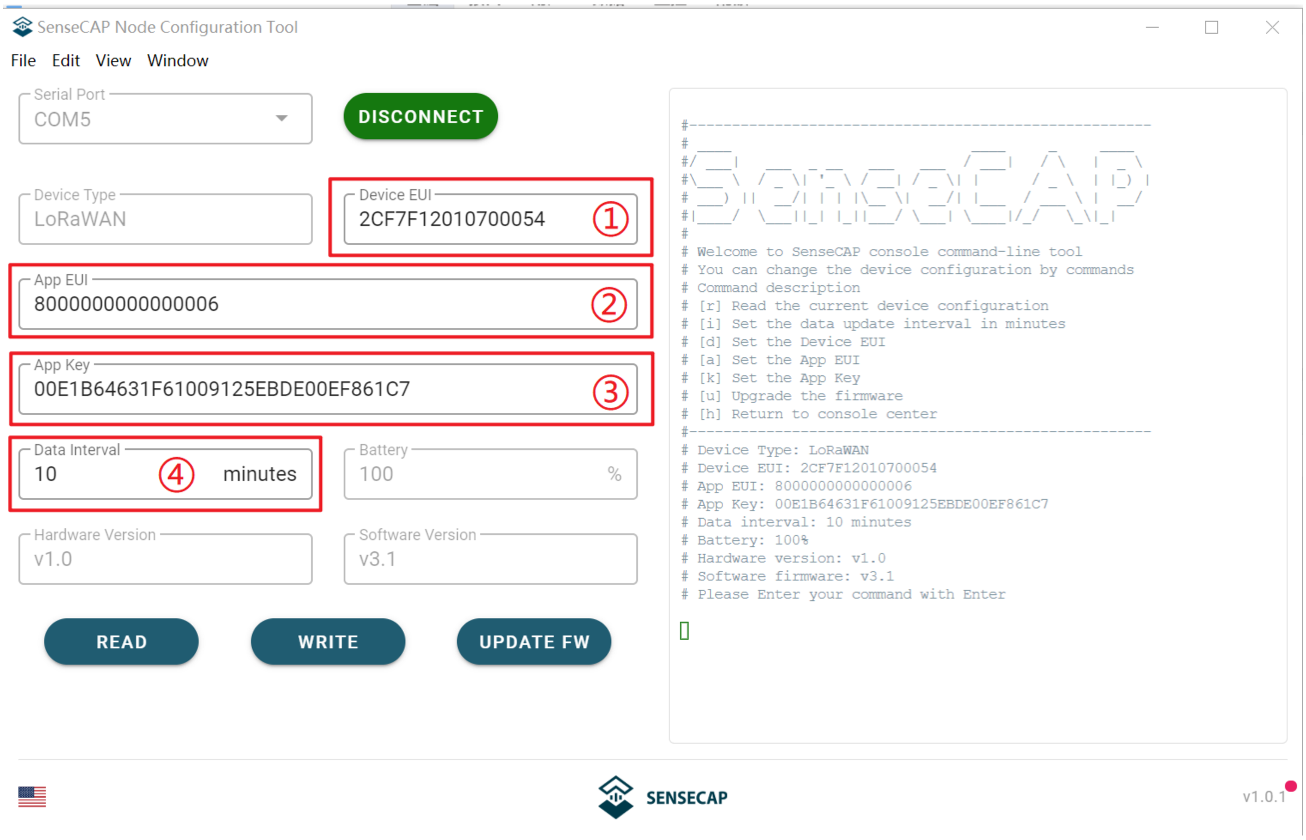
Battery: 100 %

Hardware Version: v1.0

Software Version: v3.1

```
#-----------------------------------------------------------
#  ____                           ____    _    ____
#/ ___|  ___ _ __  ___  ___     / ___|  / \  |  _ \
#\___ \ / _ \ '_ \/ __|/ _ \   | |     / _ \ | |_) |
# ___) |  __/ | | \__ \  __/   | |___ / ___ \|  __/
#|____/ \___|_| |_|___/\___|    \____/_/   \_\_|
#
# Welcome to SenseCAP console command-line tool
# You can change the device configuration by commands
# Command description
# [r] Read the current device configuration
# [i] Set the data update interval in minutes
# [d] Set the Device EUI
# [a] Set the App EUI
# [k] Set the App Key
# [u] Upgrade the firmware
# [h] Return to console center
#-----------------------------------------------------------
# Device Type: LoRaWAN
# Device EUI: 2CF7F12010700054
# App EUI: 8000000000000006
# App Key: 00E1B64631F61009125EBDE00EF861C7
# Data interval: 10 minutes
# Battery: 100%
# Hardware version: v1.0
# Software firmware: v3.1
# Please Enter your command with Enter
```

(4) ①Device EUI (16 bit) ②App EUI (16 bit) ③App Key (32 bit) ④Data Interval (Sensor collection cycle)



Serial Port: COM5

Device Type: LoRaWAN

Device EUI: 2CF7F12010700054

App EUI: 8000000000000006

App Key: 00E1B64631F61009125EBDE00EF861C7

Data Interval: 10 minutes

Battery: 100 %

Hardware Version: v1.0

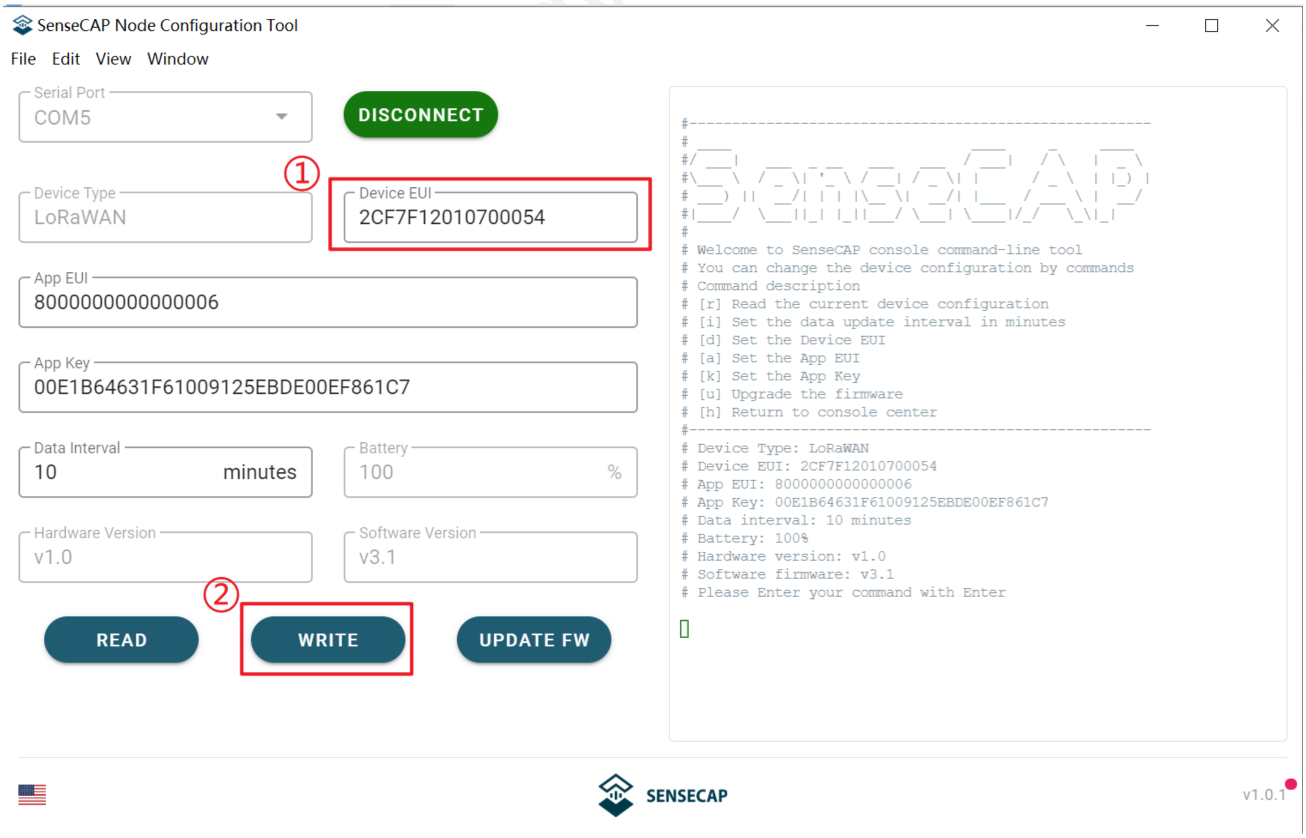
Software Version: v3.1

```
#-------------------------------------------------------------
#
#  ____                       ____    _    ____
# / ___|  ___ _ __  ___  ___ / ___|  / \  |  _ \
# \___ \ / _ \ '_ \/ __|/ _ \ |     / _ \ | |_) |
#  ___) |  __/ | | \__ \  __/ |___ / ___ \|  __/
# |____/ \___|_| |_|___/\___|\____/_/   \_\_|
#
# Welcome to SenseCAP console command-line tool
# You can change the device configuration by commands
# Command description
# [r] Read the current device configuration
# [i] Set the data update interval in minutes
# [d] Set the Device EUI
# [a] Set the App EUI
# [k] Set the App Key
# [u] Upgrade the firmware
# [h] Return to console center
#-------------------------------------------------------------
# Device Type: LoRaWAN
# Device EUI: 2CF7F12010700054
# App EUI: 8000000000000006
# App Key: 00E1B64631F61009125EBDE00EF861C7
# Data interval: 10 minutes
# Battery: 100%
# Hardware version: v1.0
# Software firmware: v3.1
# Please Enter your command with Enter
```

(5) For example: modify the Device EUI

① Write the new Device EUI.

② Click “WRITE”



Serial Port: COM5

Device Type: LoRaWAN

Device EUI: 2CF7F12010700054

App EUI: 8000000000000006

App Key: 00E1B64631F61009125EBDE00EF861C7

Data Interval: 10 minutes

Battery: 100 %

Hardware Version: v1.0

Software Version: v3.1

```
#-------------------------------------------------------------
#
#  ____                       ____    _    ____
# / ___|  ___ _ __  ___  ___ / ___|  / \  |  _ \
# \___ \ / _ \ '_ \/ __|/ _ \ |     / _ \ | |_) |
#  ___) |  __/ | | \__ \  __/ |___ / ___ \|  __/
# |____/ \___|_| |_|___/\___|\____/_/   \_\_|
#
# Welcome to SenseCAP console command-line tool
# You can change the device configuration by commands
# Command description
# [r] Read the current device configuration
# [i] Set the data update interval in minutes
# [d] Set the Device EUI
# [a] Set the App EUI
# [k] Set the App Key
# [u] Upgrade the firmware
# [h] Return to console center
#-------------------------------------------------------------
# Device Type: LoRaWAN
# Device EUI: 2CF7F12010700054
# App EUI: 8000000000000006
# App Key: 00E1B64631F61009125EBDE00EF861C7
# Data interval: 10 minutes
# Battery: 100%
# Hardware version: v1.0
# Software firmware: v3.1
# Please Enter your command with Enter
```

(6) The Main Menu shows up, with respective commands. (Use other Serial Port Tool)

```
# [r] Read the current device configuration
# [i] Set the data update interval in minutes
# [d] Set the Device EUI
# [a] Set the App EUI
# [k] Set the App Key
# [u] Upgrade the firmware
# [h] Return to console center
```

6.4 Modify the Data Interval Remotely

(1) Using the Network Server's portal or API to send downlink command, then the Node will respond to the ack.

Note: The downlink command takes effect and responds the next time the node uploads data.

(2) Select Port 2, Downlink as follow:

0x00	0x89	0x00	prepareId_L	prepareId_H	duty_L	duty_H	crc-L	crc-H

0x00	Fixed field
0x89	Fixed field
0x00	Fixed field
prepareId_L	Command ID low byte, you can customize the values, it allow each command ID to be the same
prepareId_H	Command ID high byte, you can customize the values, it allow each command ID to be the same
duty_L	Data interval low byte, you can set the data interval, unit: minute
duty_H	Data interval high byte, you can set the data interval, unit: minute
crc-L	CRC low byte, it's calculated by the CRC-16/CCITT
crc-H	CRC low byte, it's calculated by the CRC-16/CCITT

(3) When you send the downlink command, the Node responds to the ack command.

0x00	0x1F	0x00	prepareId_L	prepareId_H	result	0x00	crc-L	crc-H

0x00	Fixed field
0x1F	Fixed field
0x00	Fixed field
prepareId_L	Command ID low byte, it is the same as the downlink command
prepareId_H	Command ID high byte, it is the same as the downlink command
result	If the downlink command is in force, it responds 0x01, else it responds 0x00
0x00	Fixed field
crc-L	CRC low byte, it's calculated by the CRC-16/CCITT
crc-H	CRC low byte, it's calculated by the CRC-16/CCITT

For example: Set the Node's data interval is 10 minutes.

Send the downlink command (HEX):

00 89 00 11 22 0A 00 38 B4

0x00	0x89	0x00	prepareId_L	prepareId_H	duty_L	duty_H	crc-L	crc-H
00	89	00	11	22	0A	00	38	B4

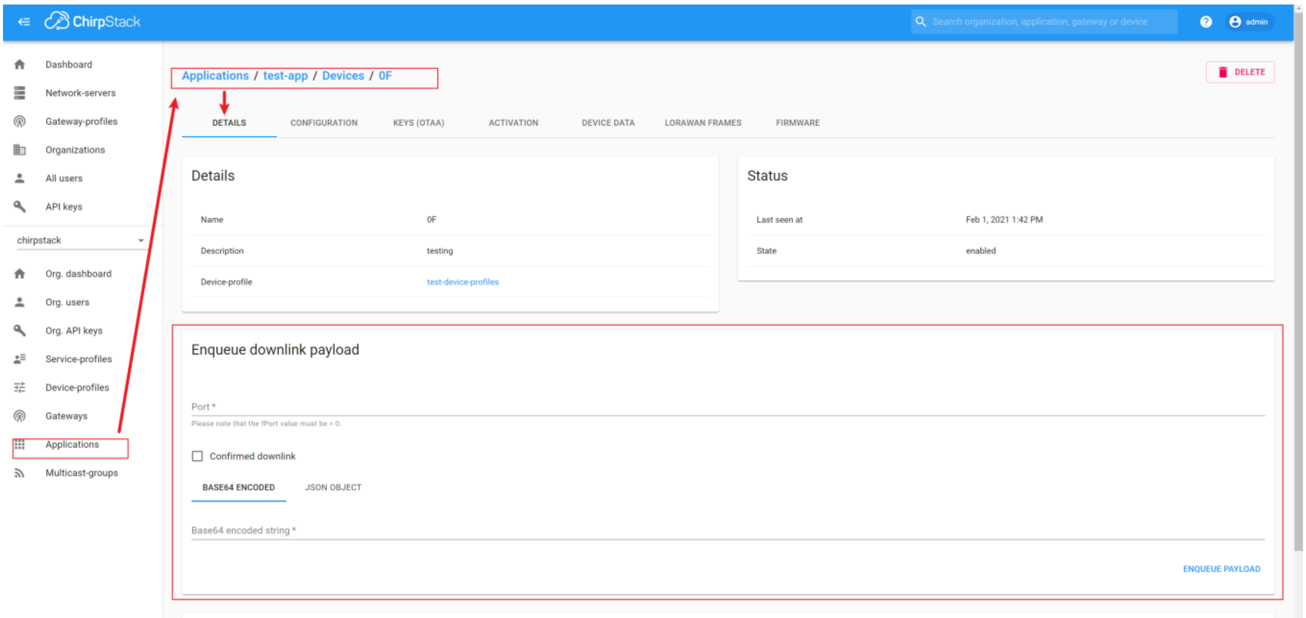
ACK Response:

00 1F 00 11 22 01 00 78 0F

0x00	0x1F	0x00	prepareId_L	prepareId_H	result	0x00	crc-L	crc-H
00	1F	00	11	22	01	00	78	0F

6.4.1 Modify the Data Interval via the Chirpstack

(1) Click to “Application→Devices→Node→DETAILS”



Details

Name	0F
Description	testing
Device-profile	test-device-profiles

Status

Last seen at	Feb 1, 2021 1:42 PM
State	enabled

(2) Enqueue downlink payload:

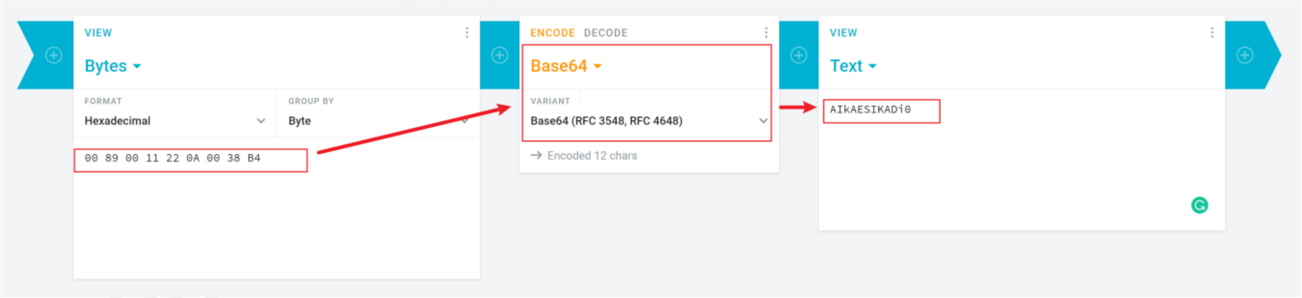
a) Port: 2

b) Select “Confirmed downlink”.

c) Input the Base64 command,

Set the Node’s data interval is 10 minutes, and send the downlink command (HEX): **00 89 00 11 22 0A 00 38 B4**

Then, use a hex to base64 tool (https://cryptii.com/pipes/hex-to-base64).



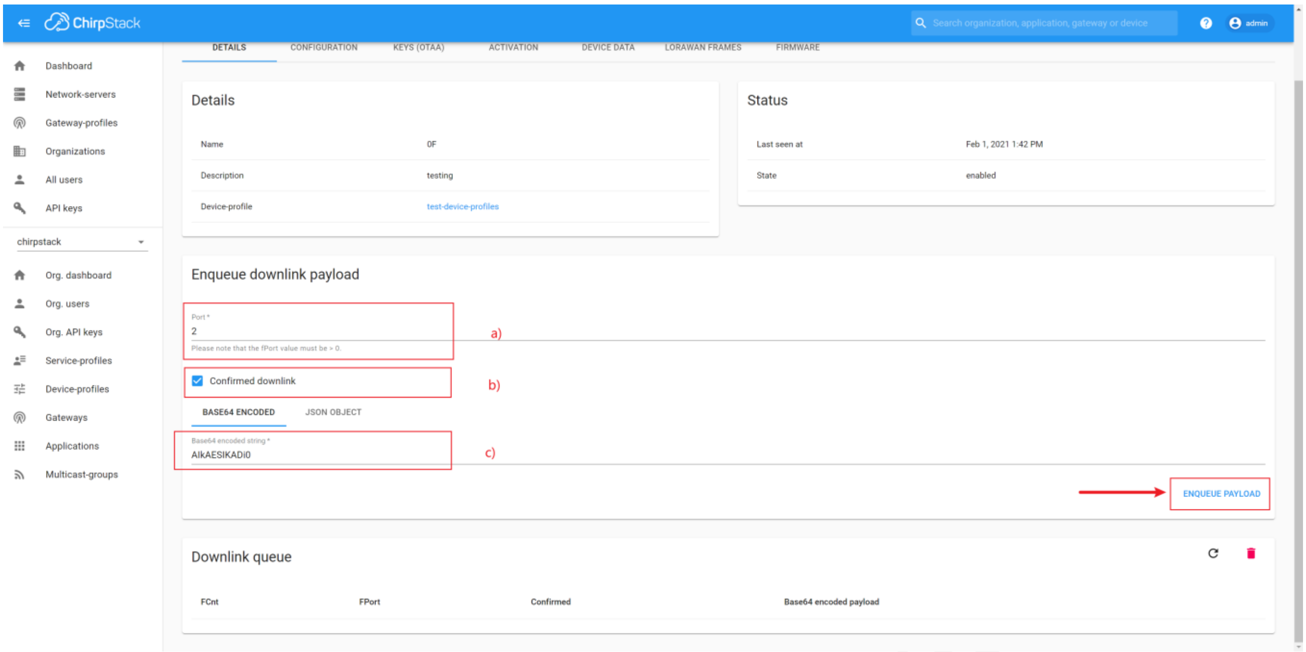
FORMAT: Hexadecimal
GROUP BY: Byte

```
00 89 00 11 22 0A 00 38 B4
```

VARIANT: Base64 (RFC 3548, RFC 4648)

```
AIkAESIKADi0
```

So, the base64 command is **AIkAESIKADi0**



Details

Name	0F
Description	testing
Device-profile	test-device-profiles

Status

Last seen at	Feb 1, 2021 1:42 PM
State	enabled

Enqueue downlink payload

Port *
2
Please note that the fPort value must be > 0.

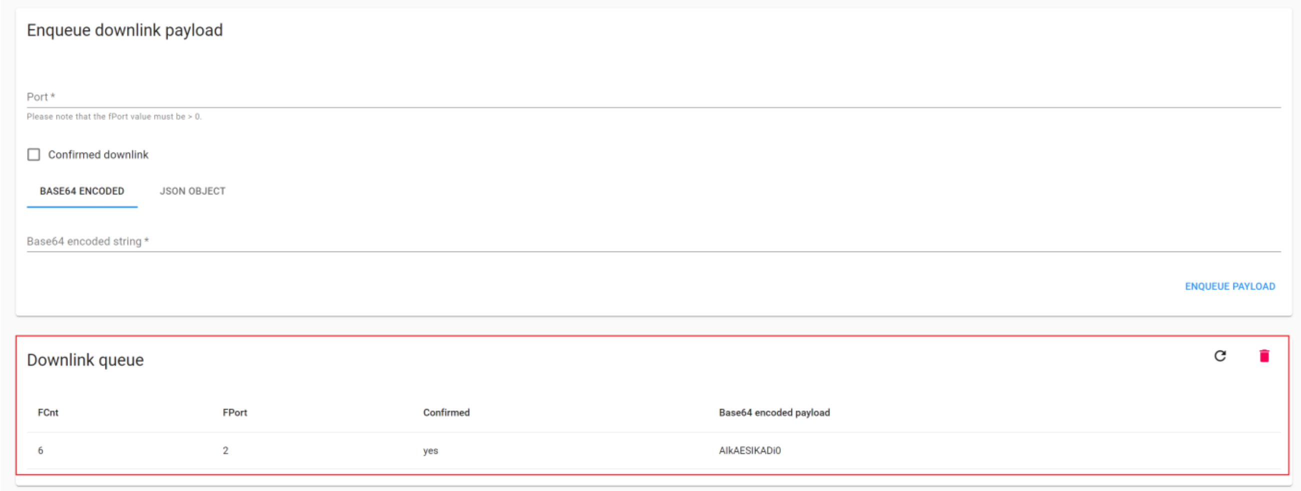
☑ Confirmed downlink

Base64 encoded string *
AIkAESIKADi0

Downlink queue

FCnt	FPort	Confirmed	Base64 encoded payload

d) Click the "ENQUEUE PAYLOAD", the "downlink queue" will display command.
When the command disappears after you refresh, the command has been sent.



Enqueue downlink payload

Port *
Please note that the fPort value must be > 0.

☐ Confirmed downlink

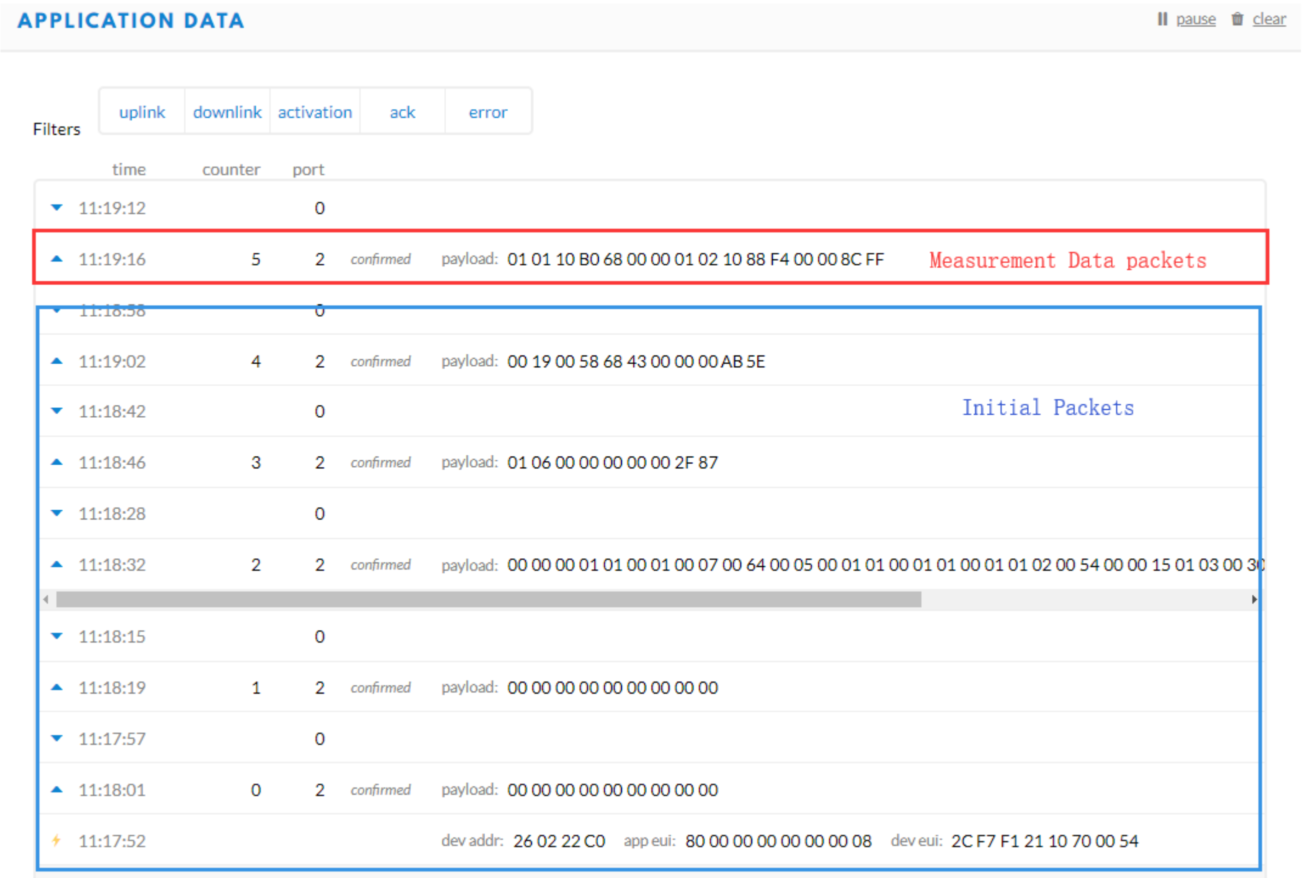
Base64 encoded string *

Downlink queue

FCnt	FPort	Confirmed	Base64 encoded payload
6	2	yes	AIkAESIKADi0

7 Decoding

In the gateway or server background, similar packets can be viewed.(If the data is encrypted, it usually needs to be decrypted using base64)



```
time      counter  port
▼ 11:19:12           0
▲ 11:19:16    5      2  confirmed  payload: 01 01 10 B0 68 00 00 01 02 10 88 F4 00 00 8C FF
▼ 11:18:38           0
▲ 11:19:02    4      2  confirmed  payload: 00 19 00 58 68 43 00 00 00 AB 5E
▼ 11:18:42           0
▲ 11:18:46    3      2  confirmed  payload: 01 06 00 00 00 00 00 2F 87
▼ 11:18:28           0
▲ 11:18:32    2      2  confirmed  payload: 00 00 00 01 01 00 01 00 07 00 64 00 05 00 01 01 00 01 01 00 01 01 02 00 54 00 00 15 01 03 00 3
▼ 11:18:15           0
▲ 11:18:19    1      2  confirmed  payload: 00 00 00 00 00 00 00 00 00
▼ 11:17:57           0
▲ 11:18:01    0      2  confirmed  payload: 00 00 00 00 00 00 00 00 00
⚡ 11:17:52                        dev addr: 26 02 22 C0   app eui: 80 00 00 00 00 00 00 08   dev eui: 2C F7 F1 21 10 70 00 54
```

Notice:

With successful access to the network, please connect the Sensor Probe back to the Sensor Node Controller by turning it clockwise. Please note the labels on both sides should be aligned as the image below, or it will not be put back in the right way. When the Sensor Probe is connected to the Sensor Node Controller correctly, the device can upload data.

7.1 Packet Parsing

Packet Initialization

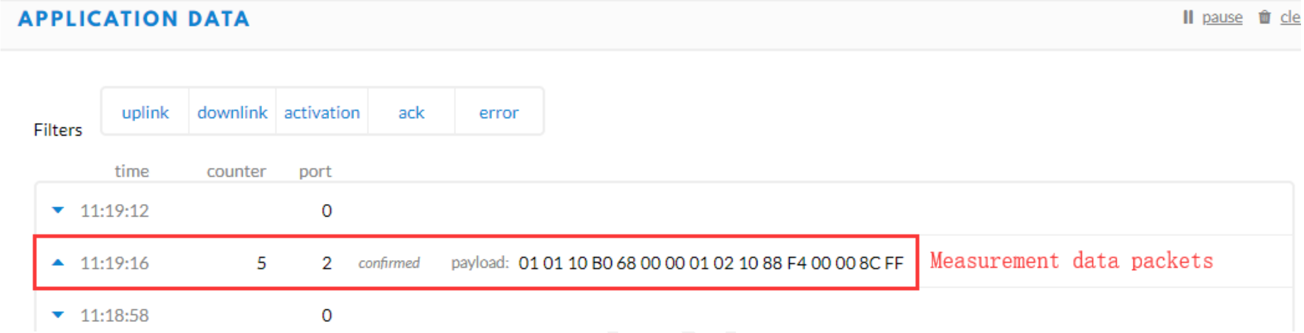
After being powered on or reboot, SenseCAP Sensor Nodes will be connected to the network using OTAA activation method. Each Sensor Node will send data packets to the server, including the following data:

Initial packets (no need to learn about these initial packets)

- One packet with device info including hardware version, software version, battery level, sensor hardware & software version, sensor EUI, power, and sensor power time counter at each channel.

Measurement data packets

The only thing we should pay attention to is the sensor measurement data packets



time	counter	port		
11:19:12		0		
11:19:16	5	2	confirmed	payload: 01 01 10 B0 68 00 00 01 02 10 88 F4 00 00 8C FF
11:18:58		0		

Measurement data packets

Packet Structure

The structure of the frame is shown in the image below.

channel	frame type	frame content
1 byte	2 bytes	≥ 4 bytes

1 byte for channel, default as 1, means the sensor has been well connected.

2 bytes for frame type, in this case, it will be 0110 and 0210, means temperature value and humidity value
4 bytes for content, is the sensor value with CRC
The frame content is sent in little-endian byte order

7.1.1 Example 1 - Air Temperature & Humidity Sensor:

Air Temperature & Humidity Sensor measurement packet: 010110B0680000010210 88F400008CFF

Divide the data into 3 sections

1	Air Temperature	010110B0680000	01 is the channel number. 0110 is 0x1001 *(little-endian byte order)* , which is the measurement ID for air temperature. B0680000 is actually 0x000068B0, whose equivalent decimal value is 26800. Divide it by 1000, and you' ll get the actual measurement value for air temperature as **26.8℃**.
2	Air Humidity	01021088F40000	0210 is 0x1002 *(little-endian byte order)* , which is the measurement ID for air humidity. 88F40000 is actually 0x0000F488, whose equivalent decimal value is 62600. Divide it by 1000, and you' ll get the actual measurement value for air humidity as 62.6%RH.
3	CRC	8CFF	The CRC verification part.

7.1.2 Example 2 - CO2 Sensor:

CO2 Sensor measurement packet: 010410E08D05009802

Divide the data into 3 sections

1	CO2	010410E08D0500	01 is the channel number. 0410 is 0x1004 *(little-endian byte order)* , which is the measurement ID for CO2.

			E08D0500 is actually 0x00058DE0, whose equivalent decimal value is 364000. Divide it by 1000, and you’ ll get the actual measurement value for CO2 as **364ppm**.
3	CRC	9802	The CRC verification part.

7.1.3 Example 3 - Soil Moisture and Temperature Sensor:

Soil Moisture and Temperature Sensor measurement packet: 010610007D0000010710725100009A21

Divide the data into 3 sections

1	Soil Temperature	010610007D0000	01 is the channel number. 0710 is 0x1007 *(little-endian byte order)* , which is the measurement ID for soil temperature. 007D0000 is actually 0x00007D00, whose equivalent decimal value is 32000. Divide it by 1000, and you’ ll get the actual measurement value for Soil Temperature as 32.0**℃**.
2	Soil Moisture	01071072510000	0710 is 0x1007 *(little-endian byte order)* , which is the measurement ID for soil moisture. 72510000 is actually 0x00005172, whose equivalent decimal value is 20850. Divide it by 1000, and you’ ll get the actual measurement value for Soil

			Moisture as 20.85%.
3	CRC	9A21	The CRC verification part.

7.1.4 Example 4 – Light Intensity Sensor:

Light Intensity Sensor measurement packet: 010310A0320000C3B6

Divide the data into 3 sections

1	Light Intensity	010310A0320000	01 is the channel number. 0310 is 0x1003 *(little-endian byte order)*, which is the measurement ID for Light Intensity. A0320000 is actually 0x000032A0, whose equivalent decimal value is 12960. Divide it by 1000, and you' ll get the actual measurement value for Light Intensity as 12.96**Lux**.
3	CRC	C3B6	The CRC verification part.

7.1.5 Example 5 – Barometric Pressure Sensor:

Barometric Pressure Sensor measurement packet: 010510284A140652B7

Divide the data into 3 sections

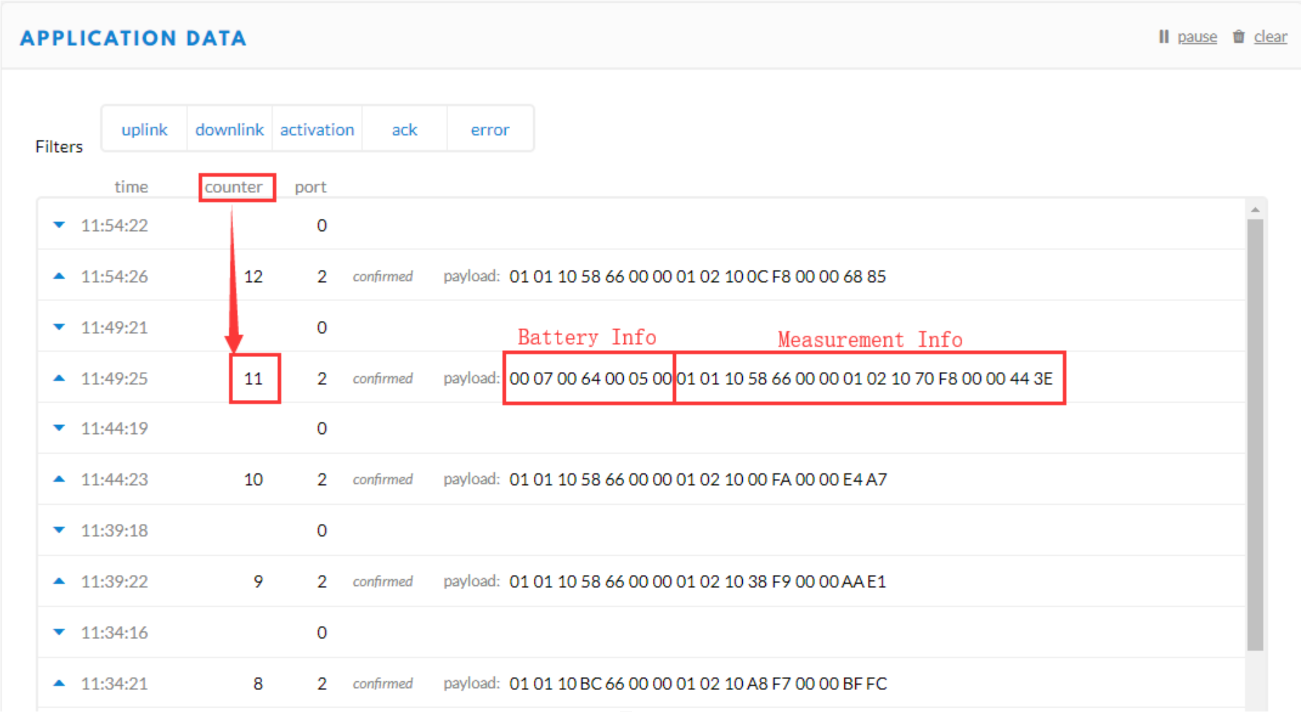
1	Barometric Pressure	010510284A1406	01 is the channel number. 0510 is 0x1003 *(little-endian byte order)*, which is the measurement ID for Barometric Pressure.

			284A1406 is actually 0x06144A28, whose equivalent decimal value is 101993000. Divide it by 1000, and you’ ll get the actual measurement value for Barometric Pressure as 101993**Pa**.
3	CRC	52B7	The CRC verification part.

To get more measurement ID, please visit https://sensecap-docs.seeed.cc/sensor_types_list.html

7.2 Exception

Please note the counter number. After 10 packets, it will follows one special packet with battery info. You can either ignore this packet or get rid of the battery info in your code.



APPLICATION DATA

Filters: uplink | downlink | activation | ack | error

	time	counter	port		
▼	11:54:22		0		
▲	11:54:26	12	2	confirmed	payload: 01 01 10 58 66 00 00 01 02 10 0C F8 00 00 68 85
▼	11:49:21		0		
▲	11:49:25	11	2	confirmed	payload: 00 07 00 64 00 05 00 01 01 10 58 66 00 00 01 02 10 70 F8 00 00 44 3E
▼	11:44:19		0		
▲	11:44:23	10	2	confirmed	payload: 01 01 10 58 66 00 00 01 02 10 00 FA 00 00 E4 A7
▼	11:39:18		0		
▲	11:39:22	9	2	confirmed	payload: 01 01 10 58 66 00 00 01 02 10 38 F9 00 00 AA E1
▼	11:34:16		0		
▲	11:34:21	8	2	confirmed	payload: 01 01 10 BC 66 00 00 01 02 10 A8 F7 00 00 BF FC

Original Info: 00070064000500010110586600000102107 0F80000443E

Battery Info: 00070064000500

Measurement Info: 010110586600000102107 0F80000443E

Example:

Battery & TH Sensor measurement packet: 00070064000500010110586600000102107 0F80000443E

Divide the data into 3 sections

1	Battery	00070064000500	
2	Temperature	01011058660000	01 is the channel number. 0110 is 0x1001 *(little-endian byte order)* , which is the measurement ID for air temperature.

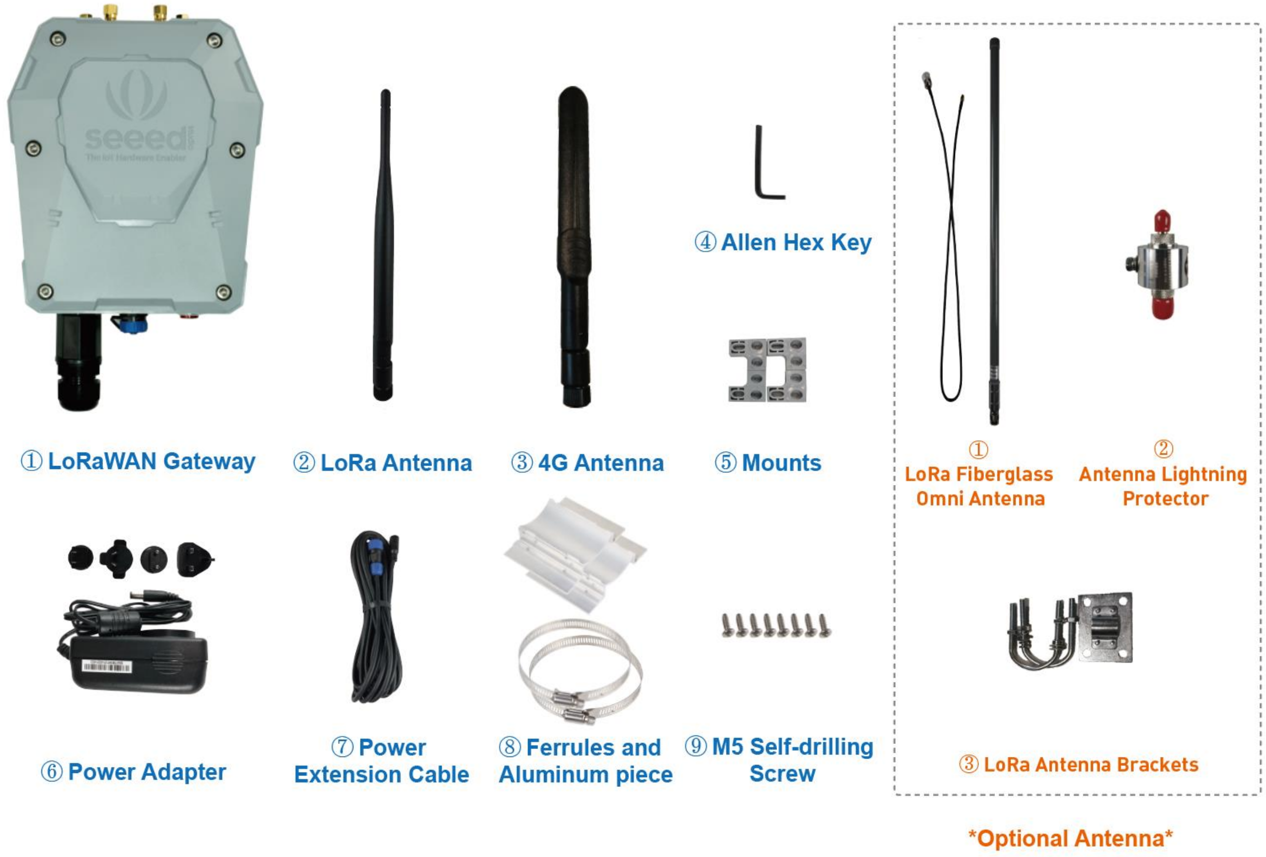
			58660000 is actually 0x00006658, whose equivalent decimal value is 26200. Divide it by 1000, and you' ll get the actual measurement value for air temperature as **26.2℃**.
2	Humidity	01021070F80000	0210 is 0x1002 *(little-endian byte order)* , which is the measurement ID for air humidity. 70F80000 is actually 0x0000F870, whose equivalent decimal value is 63600. Divide it by 1000, and you' ll get the actual measurement value for air humidity as 63.6%RH.
3	CRC	443E	The CRC verification part.

8 Device Installation

In this chapter, we will introduce the gateway and sensor nodes, their respective installation processes, as well as the dos and don'ts. Before installing, please check the part list to ensure nothing is missing.

8.1 Part List

8.1.1 Gateway Part List



The LoRa Gateway comes with a standard antenna. If you need ultra-long-distance communication, you will need to purchase a high-gain fiberglass antenna.

Item	Name	Quantity
1	LoRa Gateway	1
2	LoRa Antenna	1
3	4G Antenna	1
4	Allen Hex Key	1
5	Mounts	4
6	Power Adapter	1
7	Power Extension Cable (5M)	1
8	Ferrules / Aluminum piece	2 / 2
9	M5 Self-drilling Screw	8
10	Antenna Lightning Protector (*Optional)	1
11	LoRa Fiberglass Omni Antenna (*Optional)	1
12	LoRa Antenna Brackets (*Optional)	1

8.1.2 Sensor Node Part List

The accessories for different sensors may vary. The common parts are as follows:

Item	Name	Quantity
1	Sensor	1
2	Bracket	1
3	M4 Self-drilling Screw	4
4	M3 Self-drilling Screw	2

8.1.3 Other Accessories & Tool List

For installing in different scenarios, you might need to purchase extra accessories or tools.

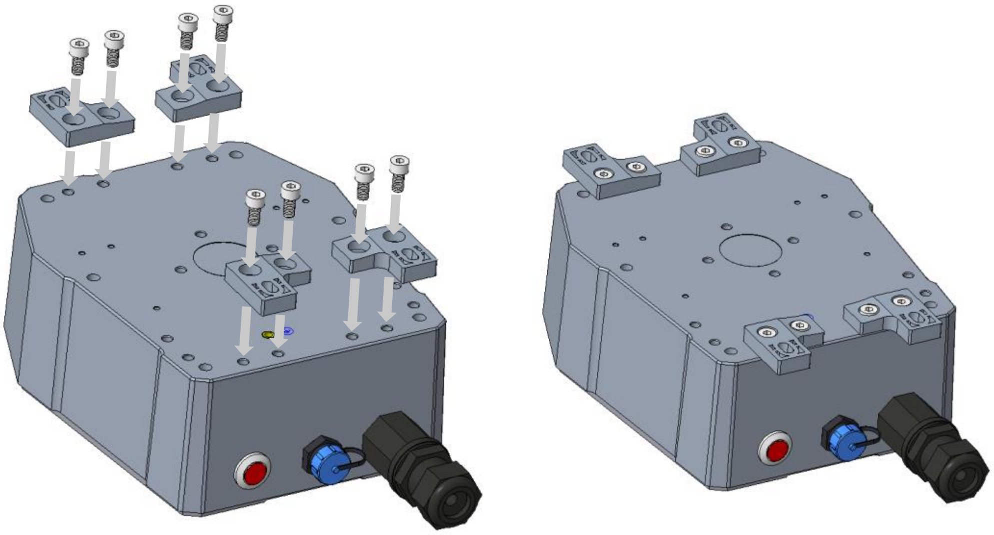
Item	Name	Quantity
1	GND Copper Wire (2.5mm^2)	2
2	Pliers	1
3	M4x12 Grounding Screw	1
4	Waterproof Self-adhesive Tape (to protect antenna connection part)	1
5	M6 Self-drilling Screw (to install the gateway on the wall)	4

8.2 Gateway Installation

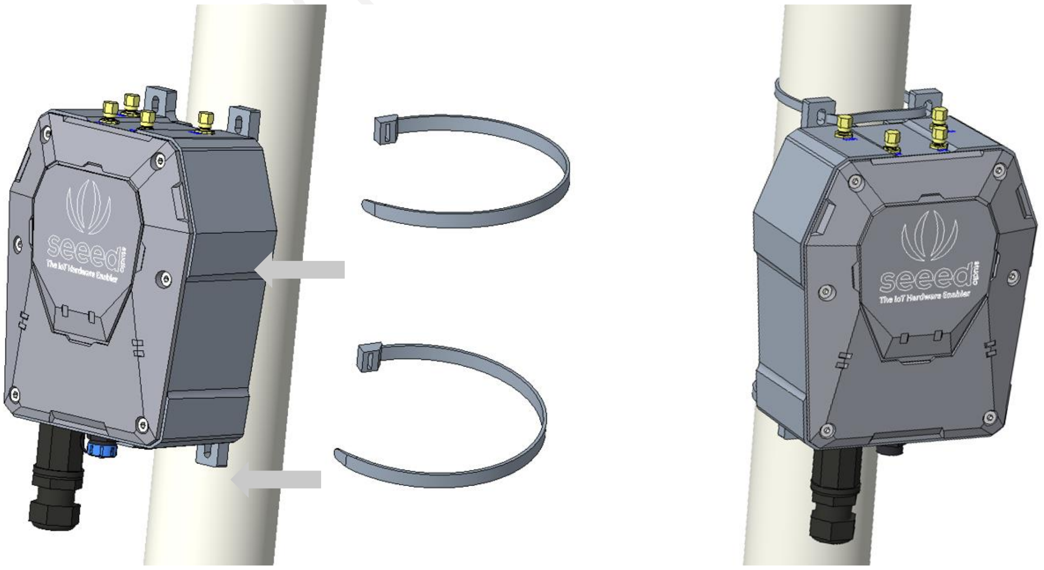
8.2.1 Gateway Installation Methods

- **Installing on a pole (Use the Mounts)**

Firstly, use M5 self-drilling screws (included in the package) to fasten the 4 brackets onto the gateway. And then use cable ties to fasten the gateway onto the pole. The recommended pole diameter is 70mm.

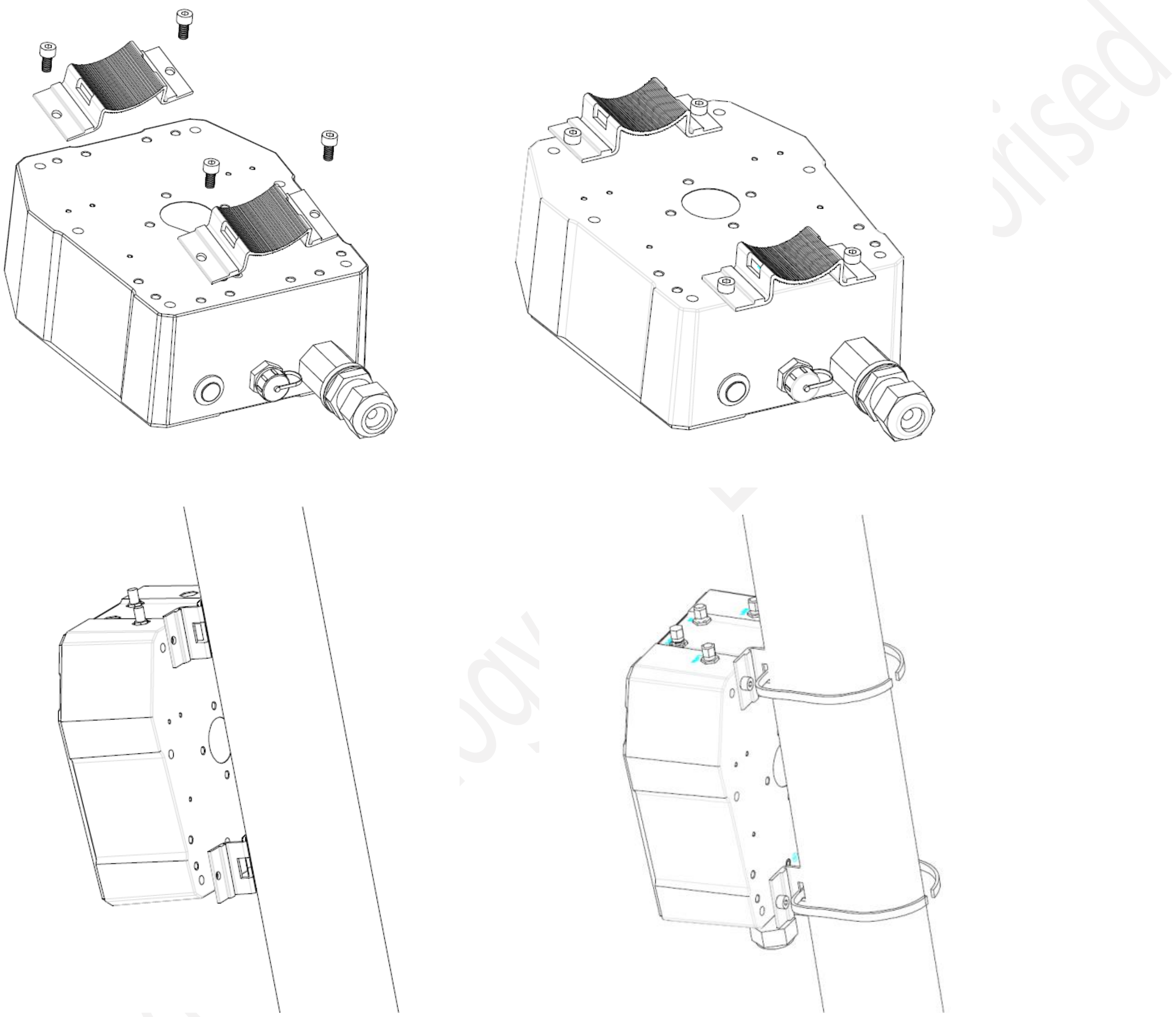


Put cable ties through the holes of the bracket and pull to fasten onto the pole. To get a better communication range, it is recommended to mount the gateway 3 meters above the ground. If there are tall buildings around, the gateway should be kept away from the building or mounted on top of the tall building.



- **Installing on a pole (Use the Ferrules and Aluminum pieces)**

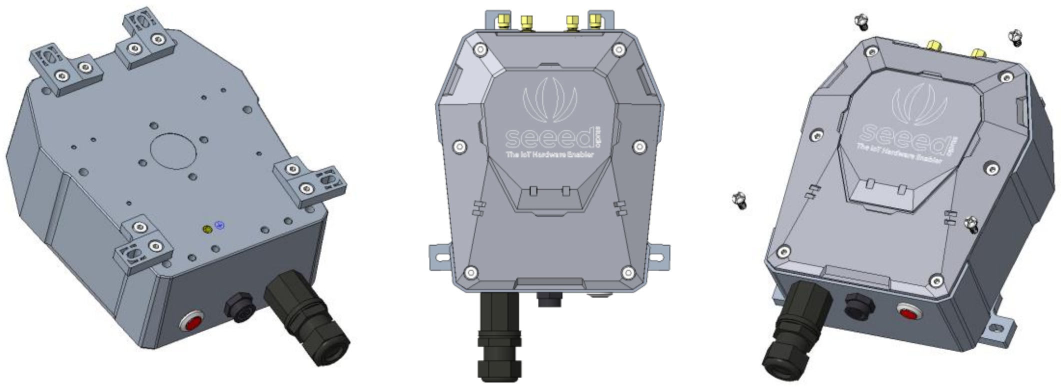
Firstly, use M5 self-drilling screws (included in the package) to fasten the 2 Aluminum pieces onto the gateway. And then use ferrules to fasten the gateway onto the pole. The recommended pole diameter is 76mm.



> **Note:** If the pole is made of metal, the antenna should be pulled higher than the metallic part of the pole, or the communication signal will have interfered.

- **Installing on the Wall**

Firstly, use M5 self-drilling screws (included) to fasten the 4 brackets onto the enclosure of the gateway (refer to the image below for directions). And then fasten the gateway onto the wall with screws.

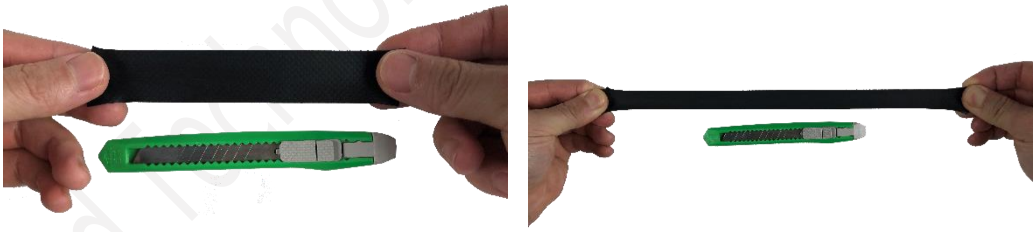


Note: The screws (that fasten gateway onto the wall) are not included in the package. Please prepare screws according to the wall materials (recommended screw diameter: 6mm).

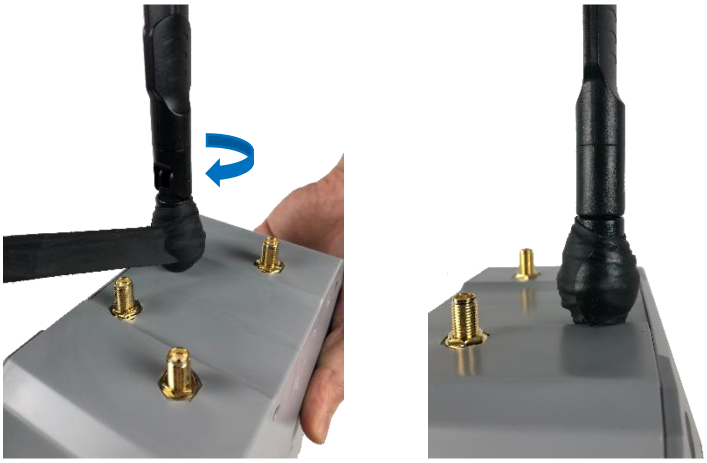
8.2.2 Installation Precautions

1) In mountainous or thunderstorm-stricken areas, please take lightening protection measures. For the fiberglass LoRa antenna, you will need to install a lightening arrester and make sure it is connected to the ground. Besides, the gateway should be mounted lower than the lightening rod.

2) When installing the gateway in the outdoor environment, the connected part should be protected with waterproof tape, to enhance waterproof performance and lengthen device lifespan. As shown below, use self-adhesive tape to protect the connection. Take a rubber tape at the length of 10cm ~ 15cm, pull it to twice of that length



wind the tape clockwise to the connected part of the antenna.



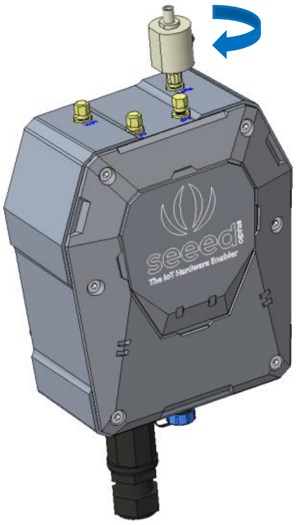
Note: The tape must be wound clockwise because the antenna is fastened clockwise. Otherwise, the antenna may loosen.

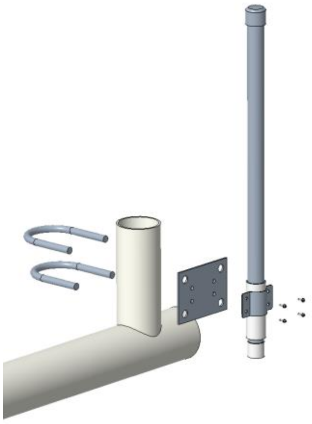
If the sensor has wires, install threaded tubes:

8.2.3 Installing Fiberglass LoRa Antenna

There are two kinds of LoRa antennas: the normal LoRa antenna (included in the package), and the fiberglass LoRa antenna (to be purchased separately). We will introduce how to install the fiberglass LoRa antenna.

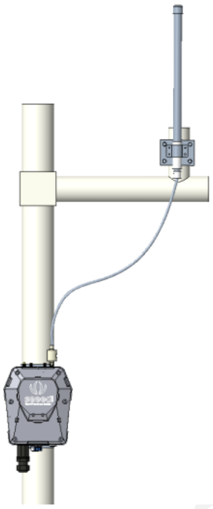
1) Fasten the lightening arrester onto the antenna port.

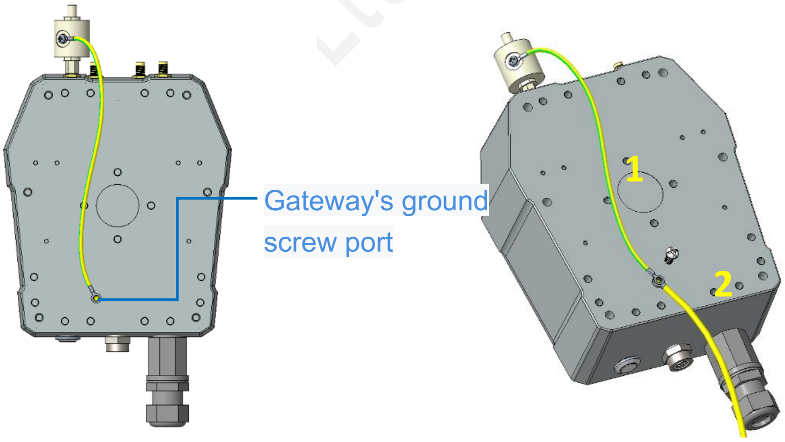




2) As shown in the image below, please fasten the fiberglass antenna onto the base part, and then fasten the whole part onto the vertical cylinder (maximum cylinder diameter: 50mm).

3) Use a 1-meter antenna feed line to connect the lightening arrester with the fiberglass antenna.





8.2.4 Installing Ground Cable

Here we will connect the lightening arrester to the GND screw port on the gateway with a ground cable, and then connect the whole device to the ground. The image below shows the location of the GND port at the backside of the gateway.

1) Prepare two copper cables, a shorter one (approx. 30cm) for connecting the lightening arrester with the GND screw port (on the gateway), and a longer one for connecting the device to the ground.

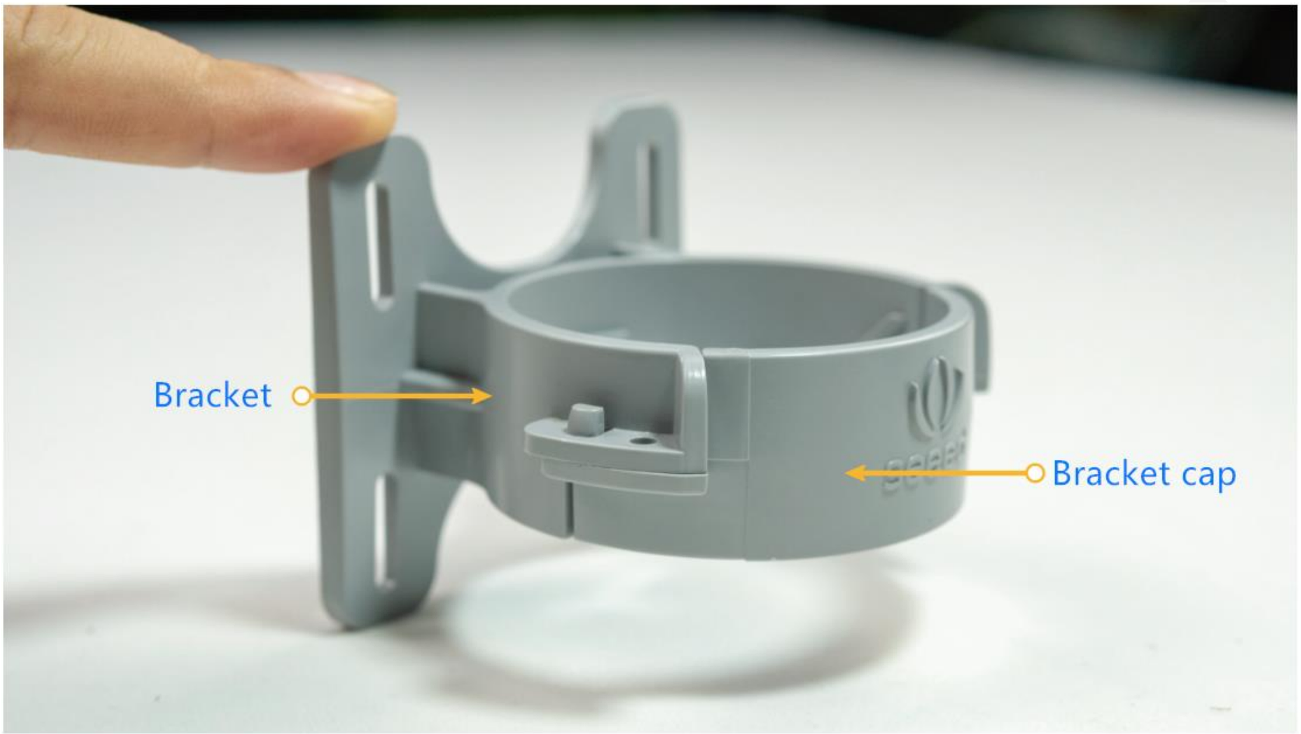
2) Fasten the lightening arrester to the short copper cable with screws, and then connect the two copper cables to the GND screw port. Use the screw to connect and fasten them.

3) Once the two cables are connected, connect the other end of the long cable to the ground. Depending on your actual installation environment, you can connect it to the ground directly or connect it to the copper ground bars.

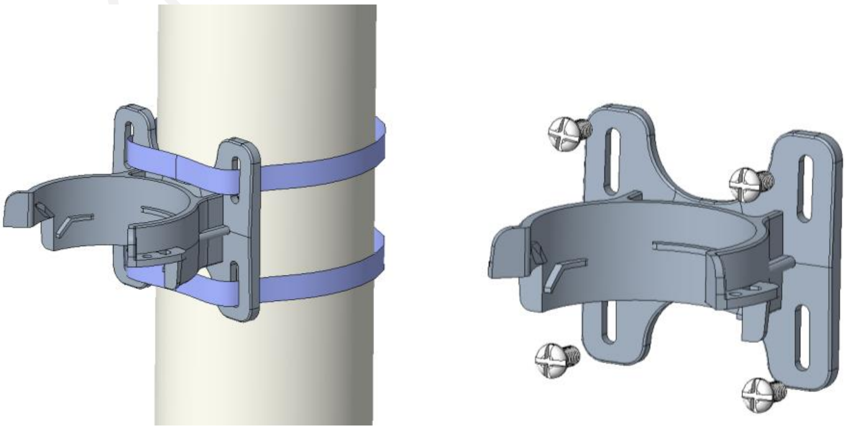
8.3 Installing Sensor Node

8.3.1 Installing the Sensor Node Bracket

Specially designed for installing SenseCAP Sensor Nodes, the bracket consists of a bracket and a sliding cap. With designated screw-holes, the bracket helps fasten the Sensor Node firmly onto a pole or a wall.



1) To install on a pole, you can use zip ties to fasten the bracket (recommended pole dimension is 50-70mm in diameter). Please refer to the following image for bracket directions.



2) To install on the wall or other surfaces, you can use self-drilling screws to fasten the bracket onto the surface.

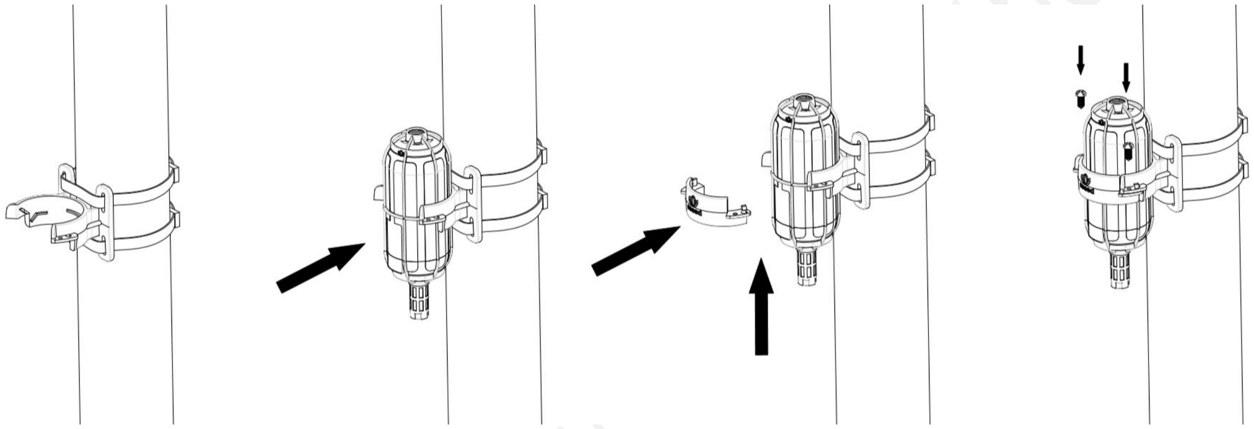
8.3.2 Installing Sensor Nodes

After installing brackets, let’s install sensors.

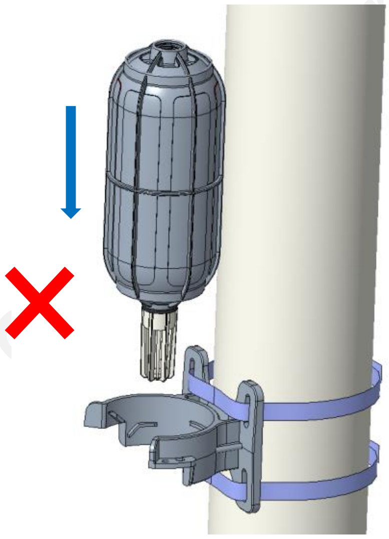
1) The Sensor Probe should be placed vertically downward with the label facing outward. Be consistent with the bracket gap. Make sure the circle part in the middle of Sensor Node is aligned with the middle of the bracket, and then press the Sensor Node to fit into the bracket. A click/snap sound indicates that the Sensor Node has been installed successfully. Try to manually twist it to make sure the Sensor Node is locked to the bracket securely.

2) Secure by fastening the bracket cap as instructed in the image.

3) Place two self-drilling screws on the bracket to increase firmness and help prevent theft.



Note: Do not insert the Sensor Node into the bracket from the top, or it will not fasten the onto the bracket securely.

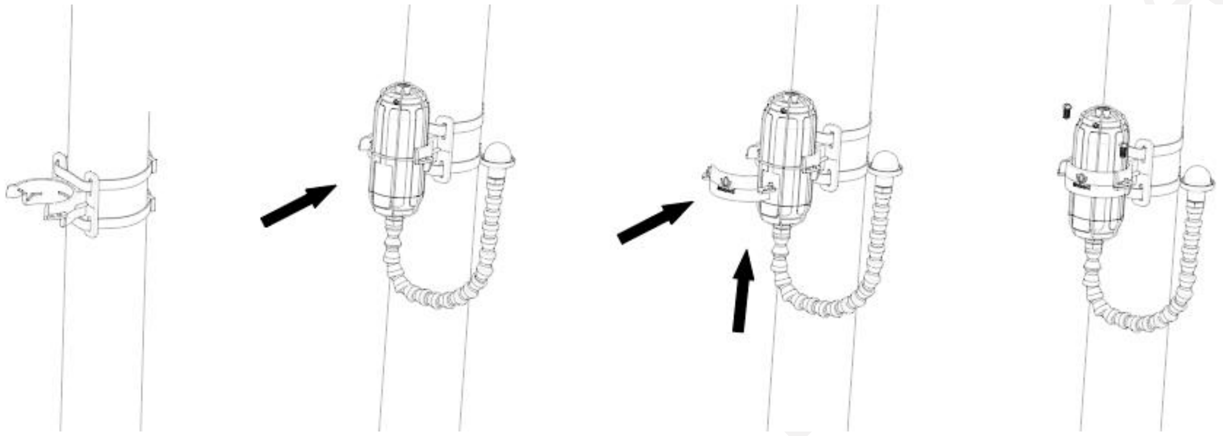


8.3.3 Dos and Don’ts in Installing Sensor Probes

The same instruction applies to installing the different Sensor Nodes. However, there are some tips to keep in mind when installing certain Sensor Nodes.

- **Light Sensor**

  The Sensor Probe of the Light Sensor needs to be placed vertically upward, and there should not be anything obstructing sunlight from the Sensor Probe.



- **CO2 Sensor**

  The Sensor Probe can be fastened with self-drilling screws. Please refer to the image below for the probe direction. The end without the cables should point downward to prevent rain or dust from getting into the probe. Also, the device should be in a place with good ventilation.