SENSECAP

LoRaWAN Gateway User Guide

Version: V1.4

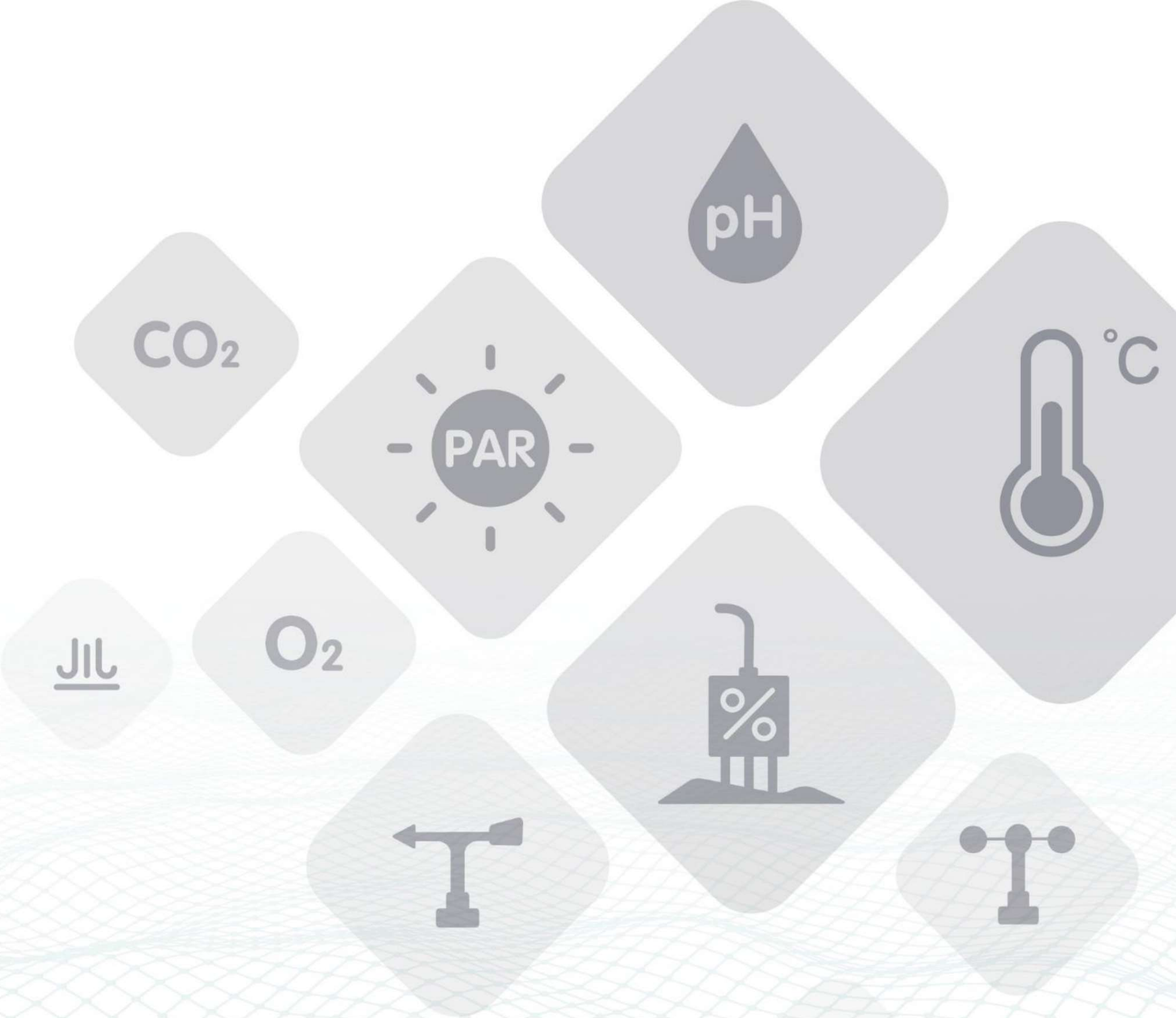


Table of Contents

1 Product Introduction

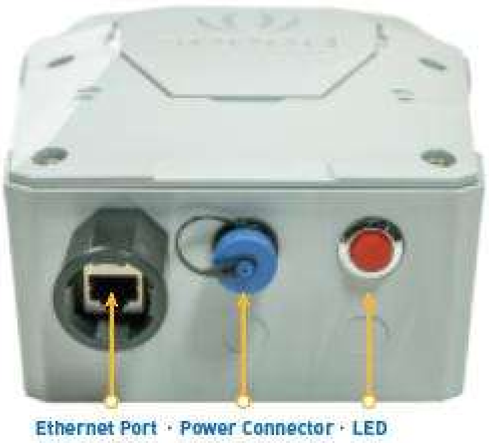
SenseCAP is an industrial wireless sensor network that integrates easy-to-deploy hardware and data API services, enabling low-power, long-distance environmental data collection. SenseCAP includes several versions, such as LoRaWAN, LoRaPP, etc.

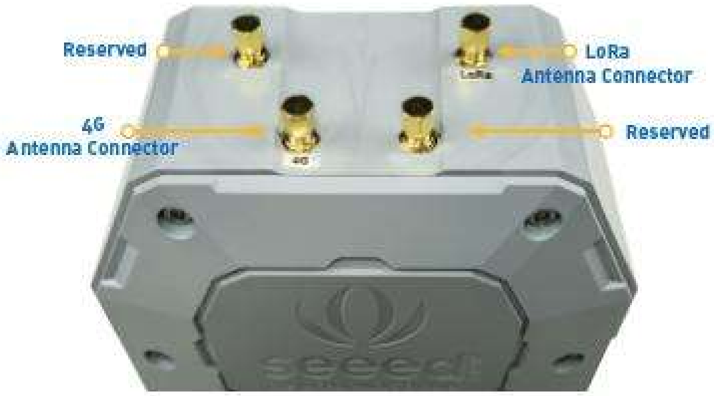
SenseCAP LoRaWAN Gateways is based on the LoRaWAN protocol, it can realize one-to-many, long-distance networking and bilateral communication. The LoRaWAN Gateway supports Ethernet and 4G.

Main Features:

- High-performance Cortex A8 1GHz processor
- Multiple methods to connect to the Internet: 4G, Wi-Fi and Ethernet
- Supports third-party TTN account and server
- 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 Outdoor Gateway:



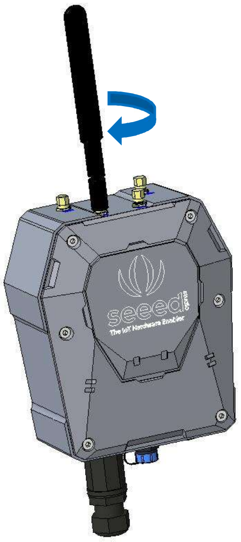


2 Gateway Network Configuration

2.1 The gateway connects to the Internet

2.1.1 Installing Antenna

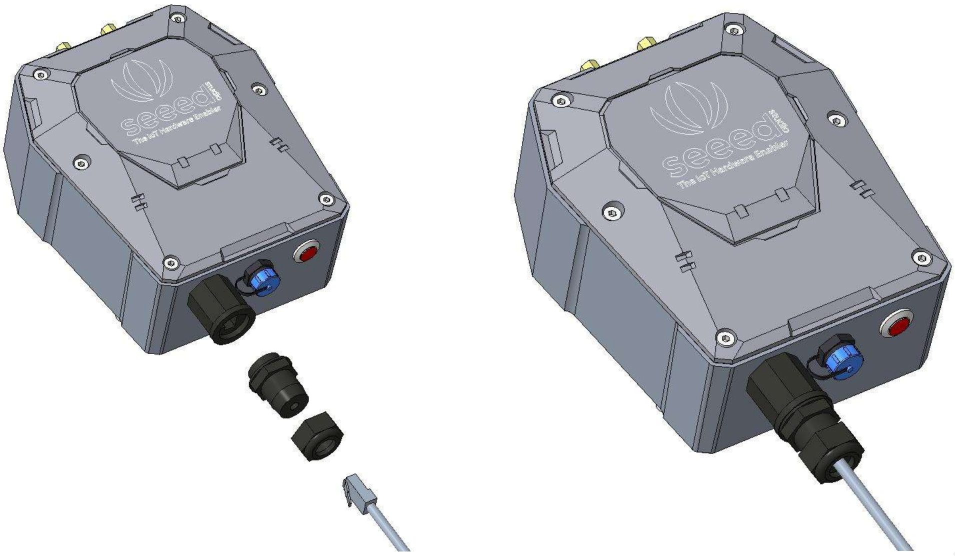
Screw clockwise to install the 4G and LoRa antennas onto the gateway.



2.1.2 Connecting to the Internet

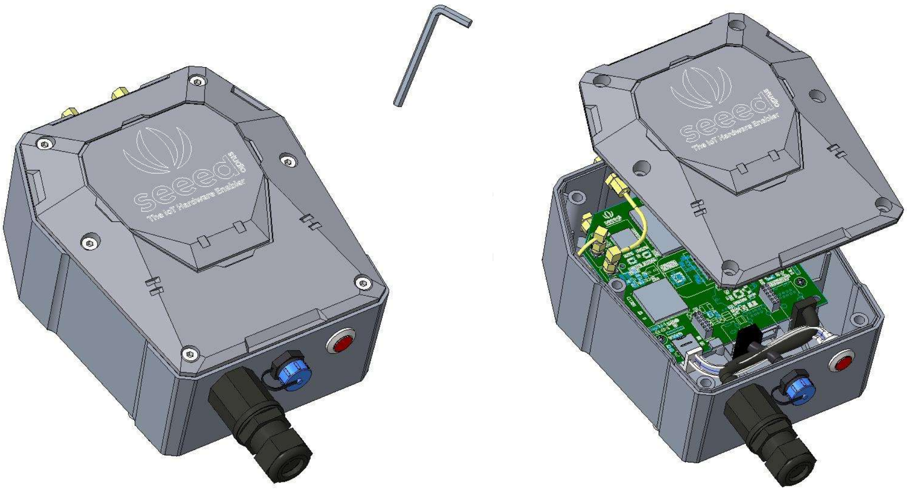
There are two ways to connect to the Internet. Choose the one that works for you .

(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.

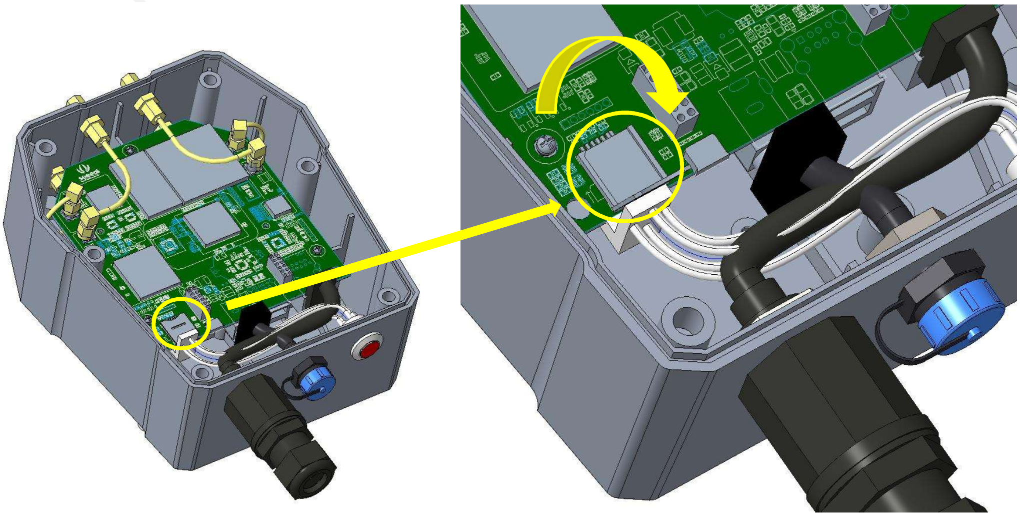


(2) Connecting to 4G

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

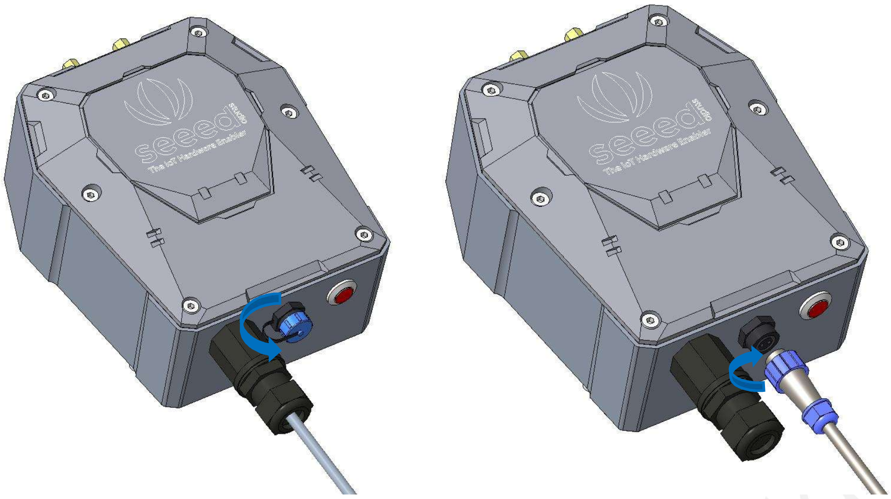


Swipe downward to open the SIM card socket, insert the Micro SIM card and swipe upward to lock the SIM card socket. Make sure it is installed correctly and close the lid with the screws.



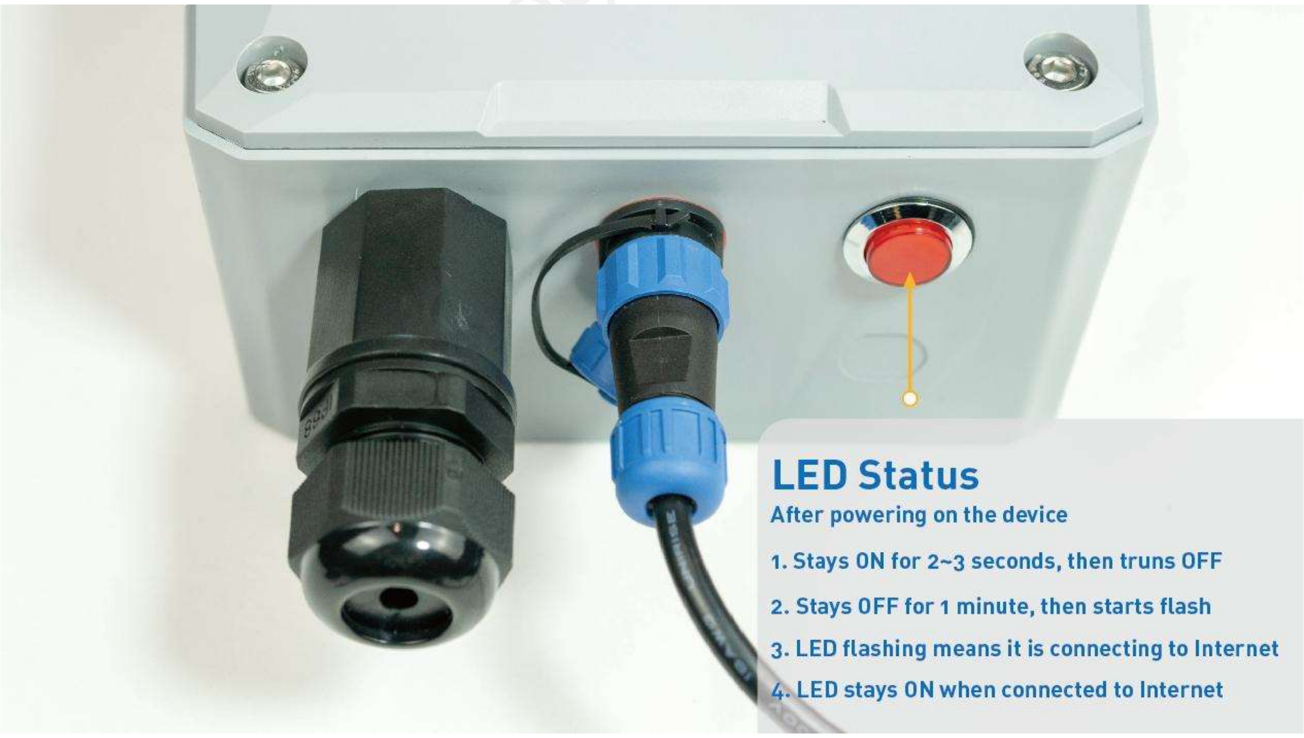
2.1.3 Connecting to Power Cable

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



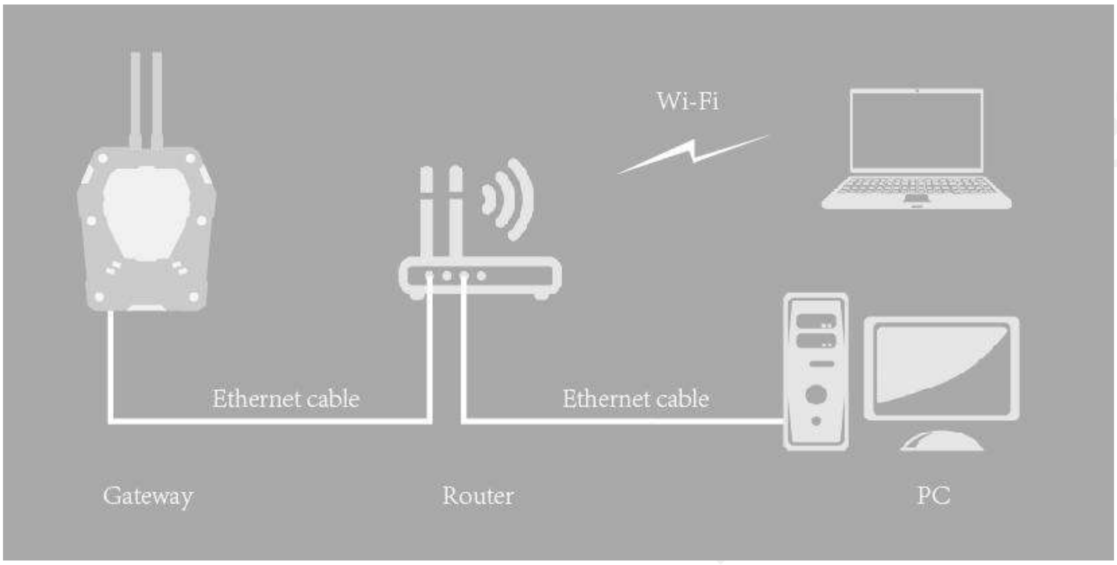
LED Status

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

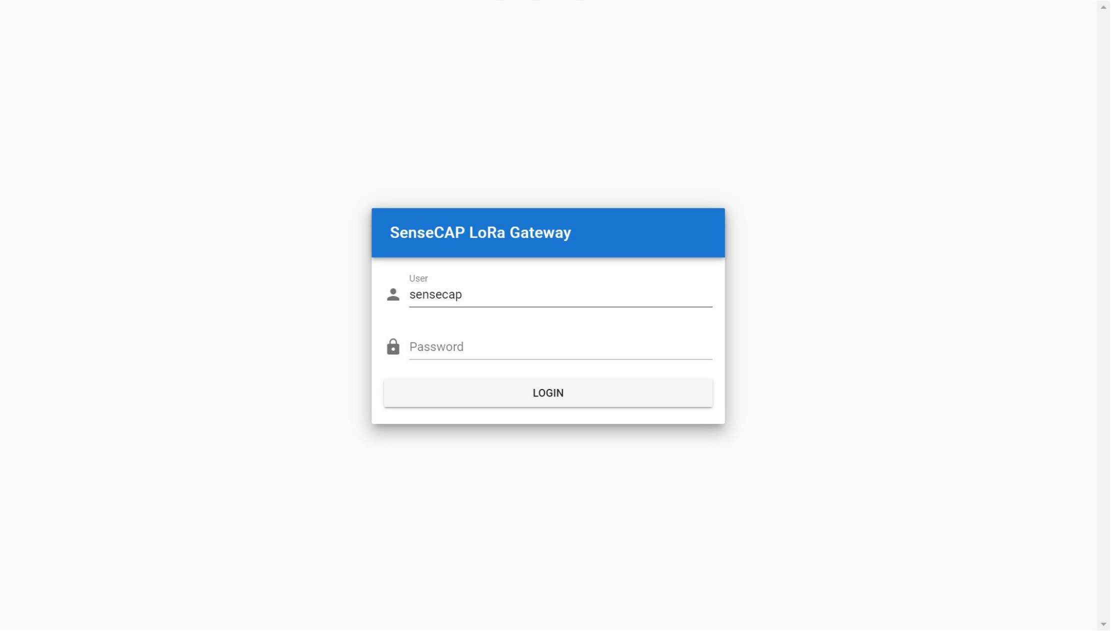
2.2 Setting the APN

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

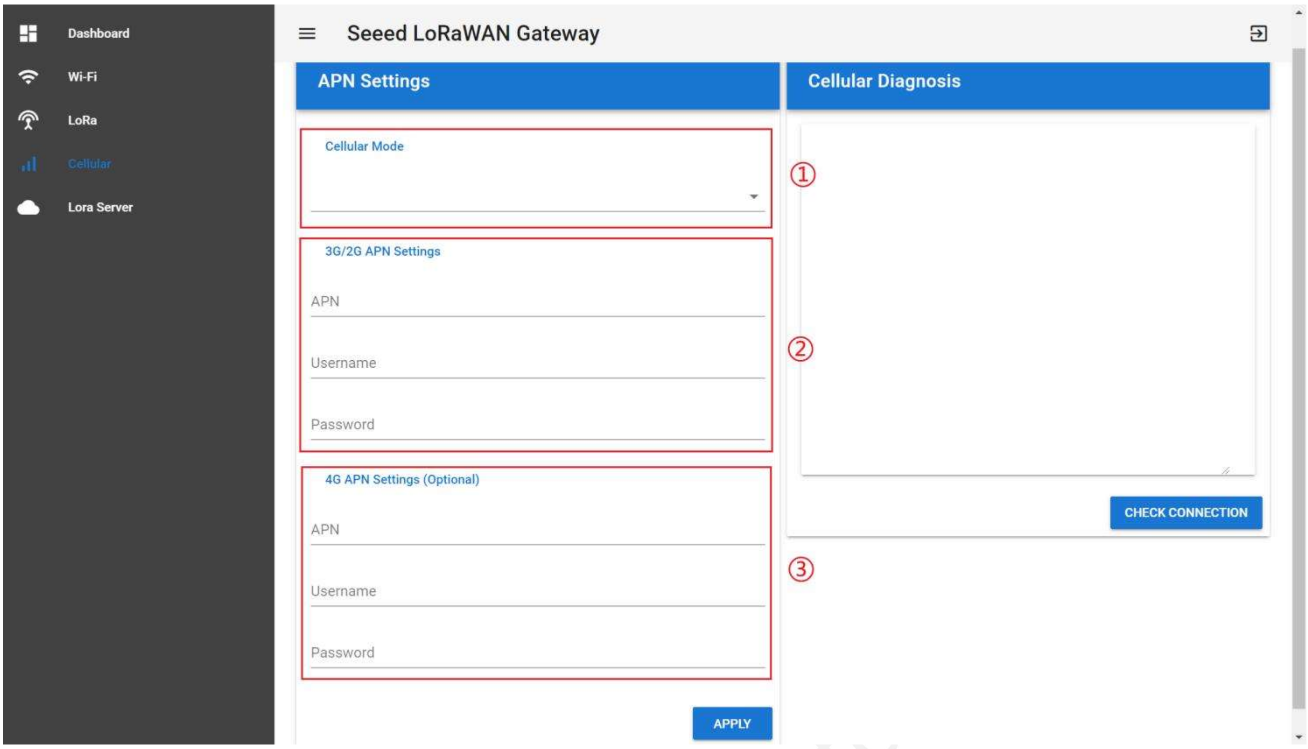

SenseCAP LoRa Gateway

User
sensecap

Password

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

(4) Click the “Cellular” button.

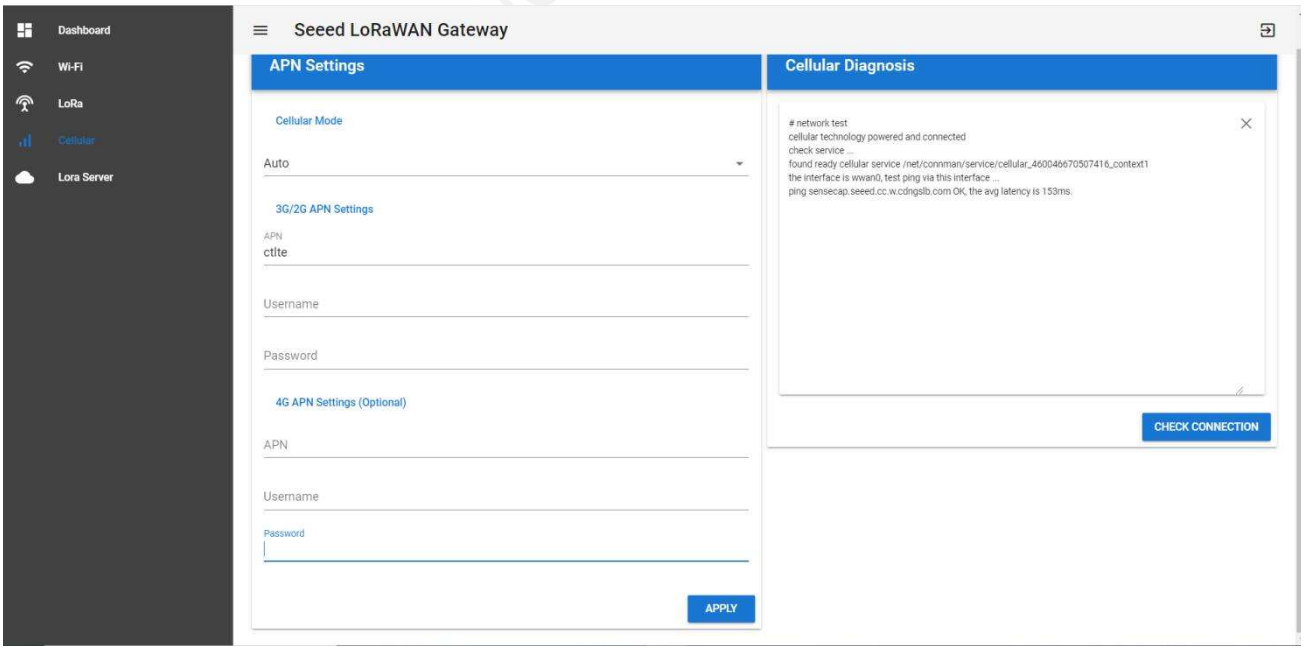


① Cellular Mode: AUTO(default), Gateway automatically selects mode.

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

3G/2G APN Settings

APN: ctlte

Username

Password

4G APN Settings (Optional)

APN

Username

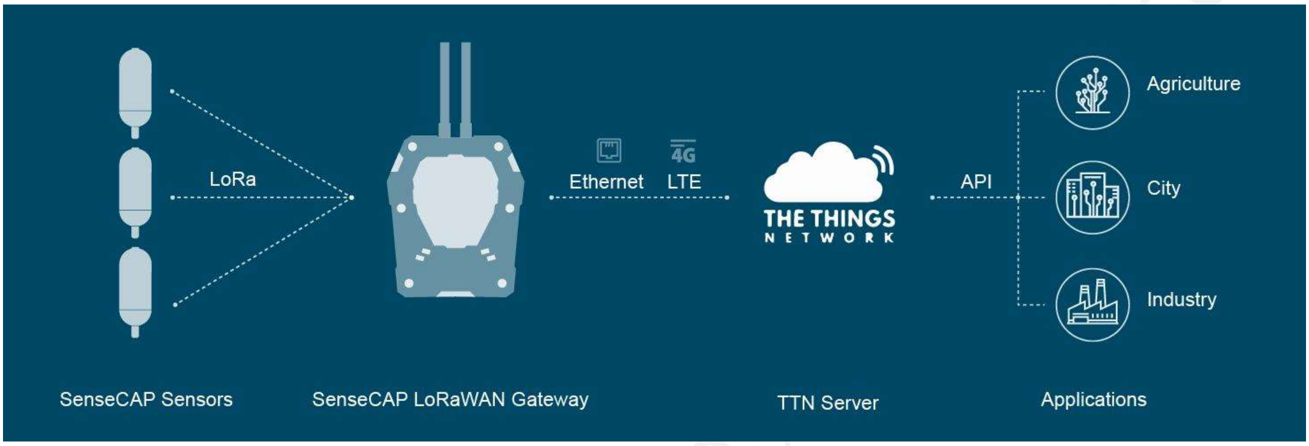
Password

```
# 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.

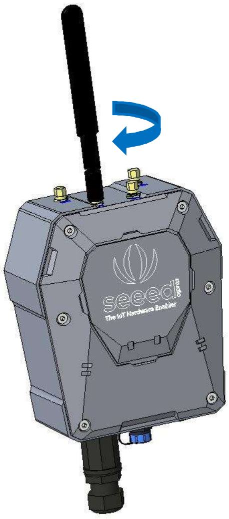
Learn more about TTN: https://www.thethingsindustries.com/docs/



3.1 Gateway Network Configuration

3.1.1 Installing Antenna

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

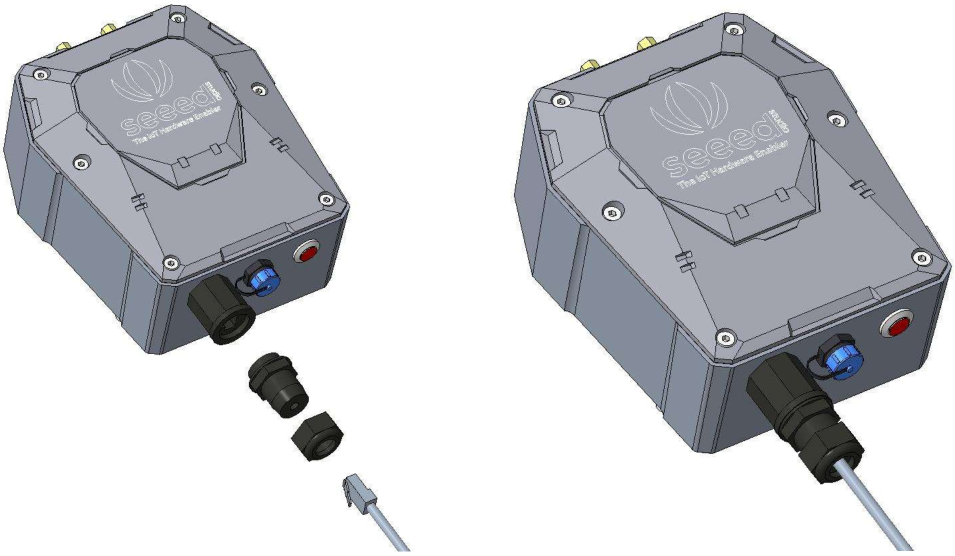


3.1.2 Connecting to the Internet

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

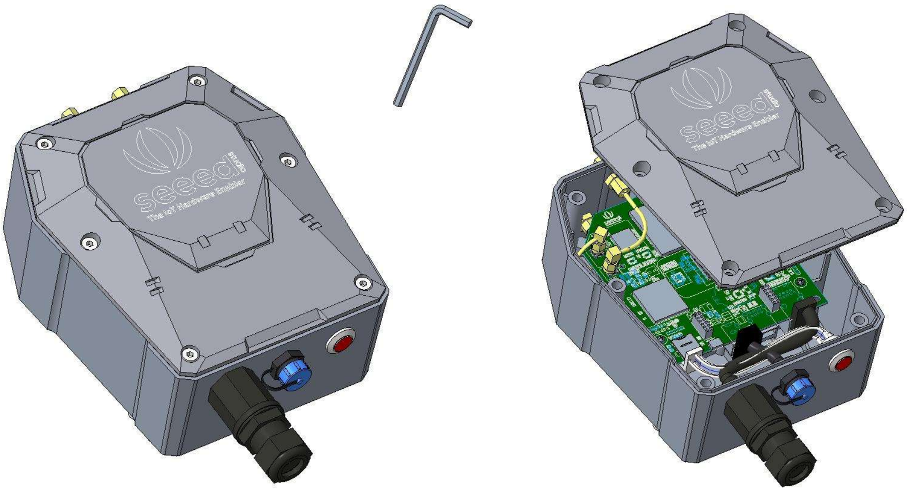
(3) Connecting to Ethernet Cable

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

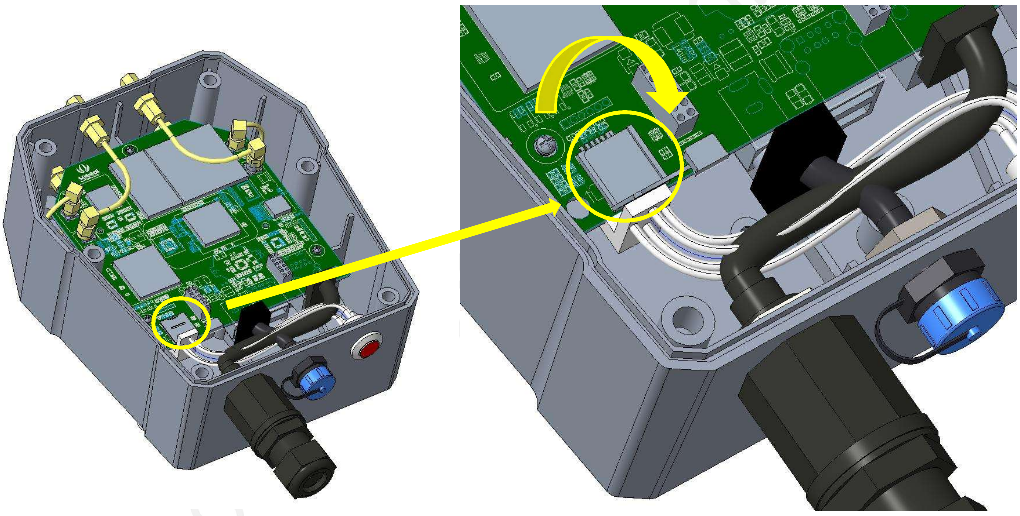


(4) Connecting to 4G

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

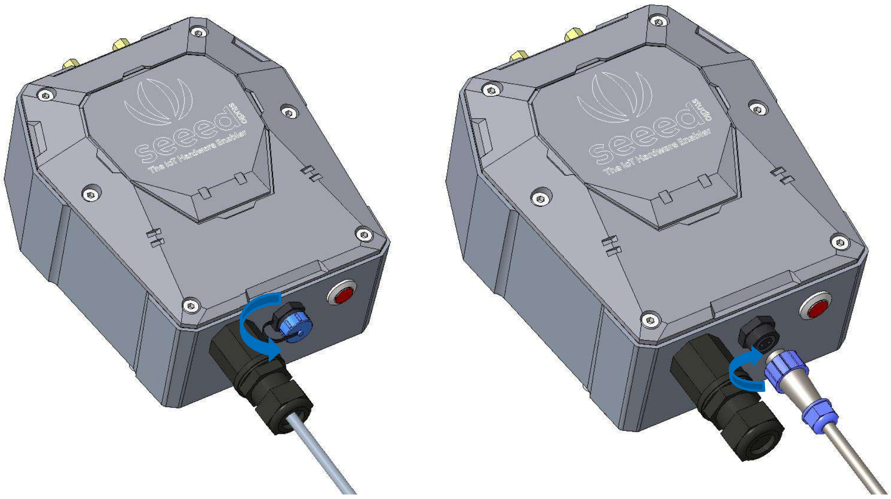


Swipe downward to open the SIM card socket, insert the Micro SIM card and swipe upward to lock the SIM card socket. Make sure it is installed correctly and close the lid with the screws.



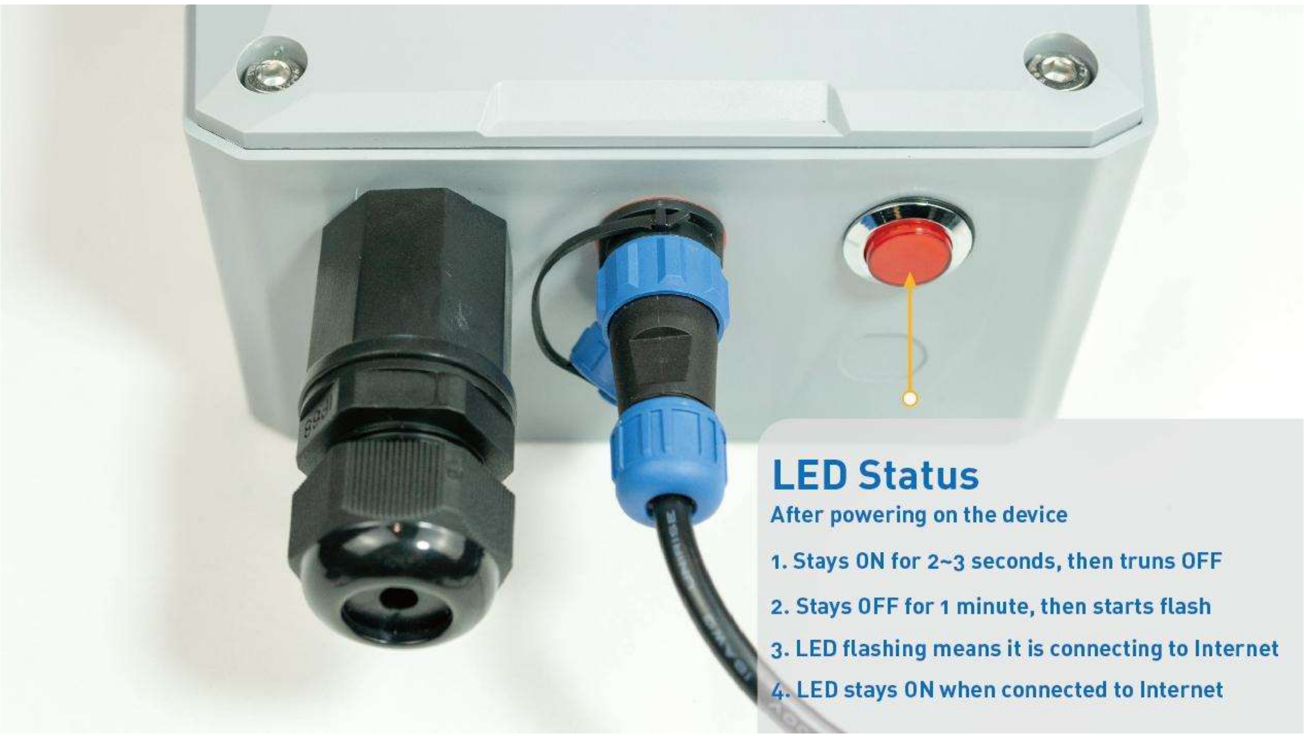
3.1.3 Connecting to Power Cable

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



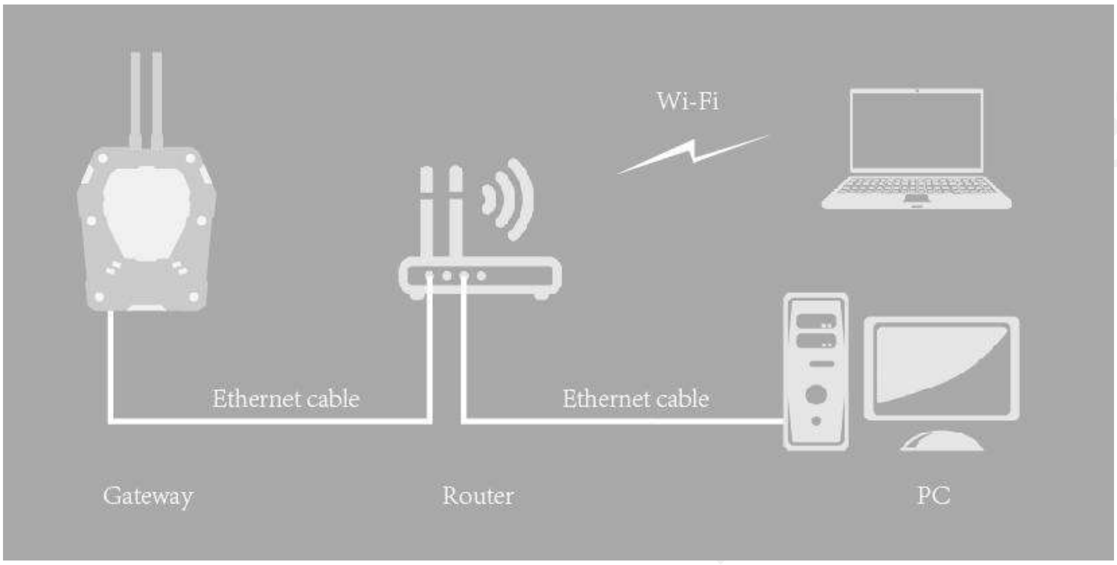
LED Status

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

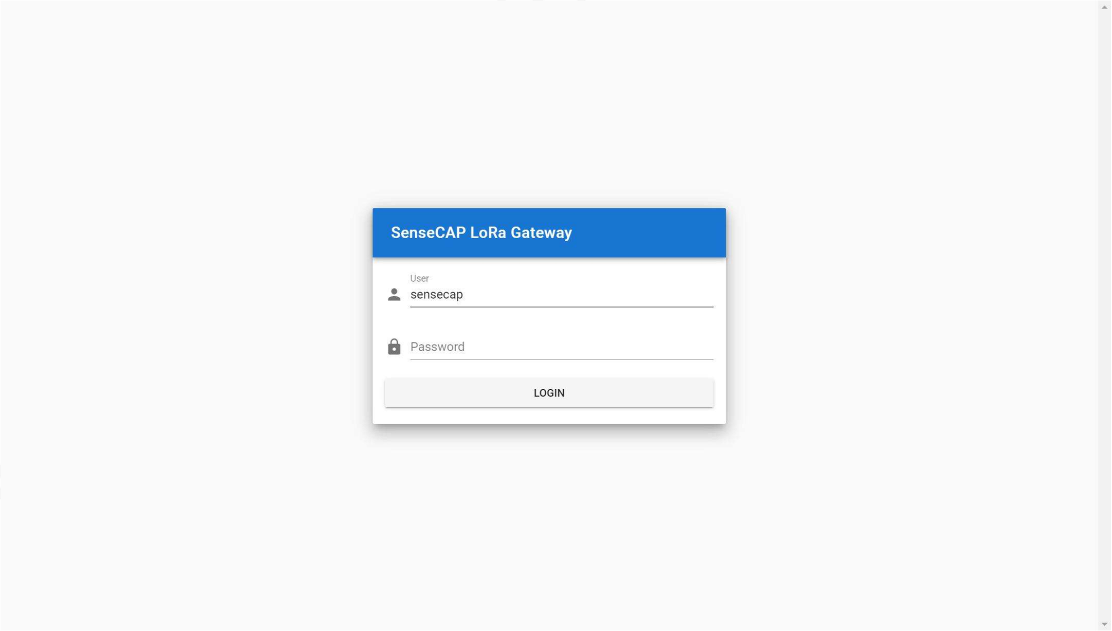
3.2 Setting the Gateway Service Address

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

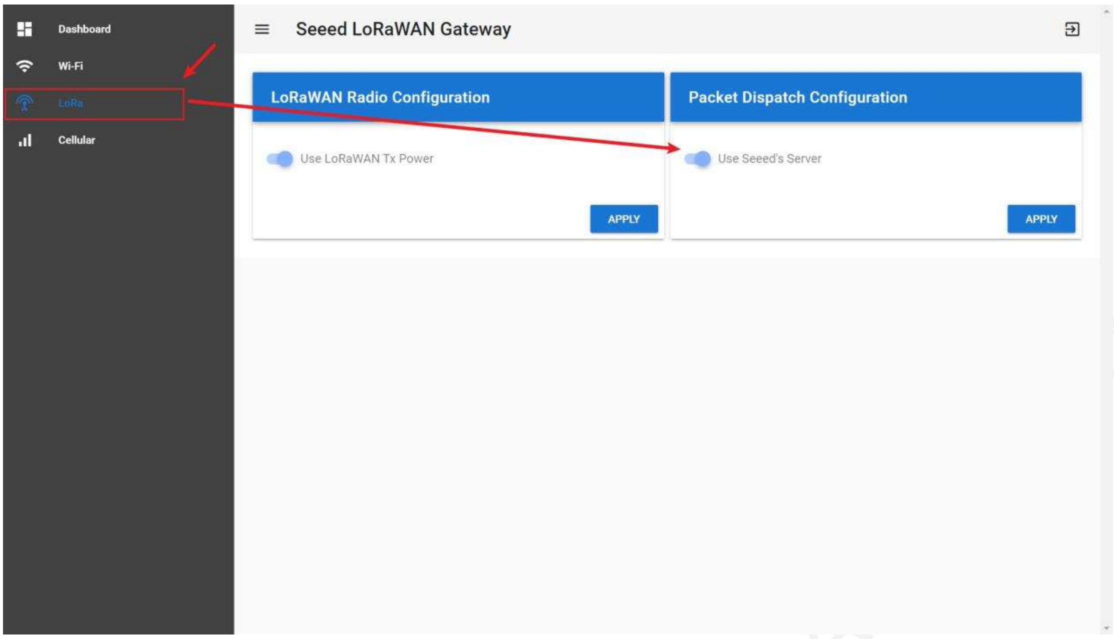

SenseCAP LoRa Gateway

User
sensecap

Password

(8) User: sensecap
Password: sensecap!!!

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



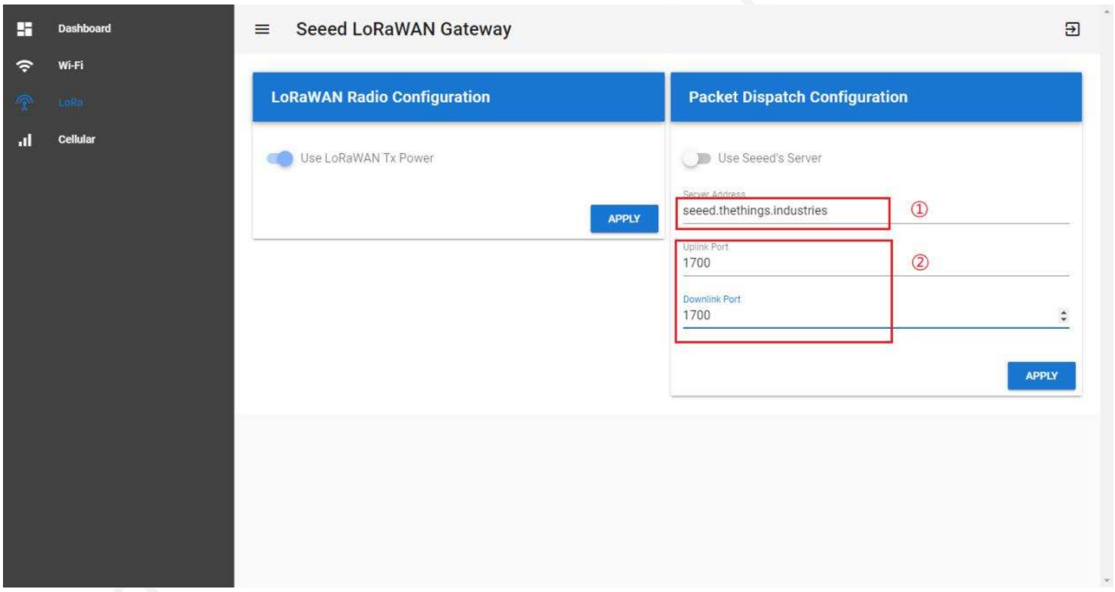
LoRaWAN Radio Configuration

Use LoRaWAN Tx Power

Packet Dispatch Configuration

Use Seeed's Server

(10)



LoRaWAN Radio Configuration

Use LoRaWAN Tx Power

Packet Dispatch Configuration

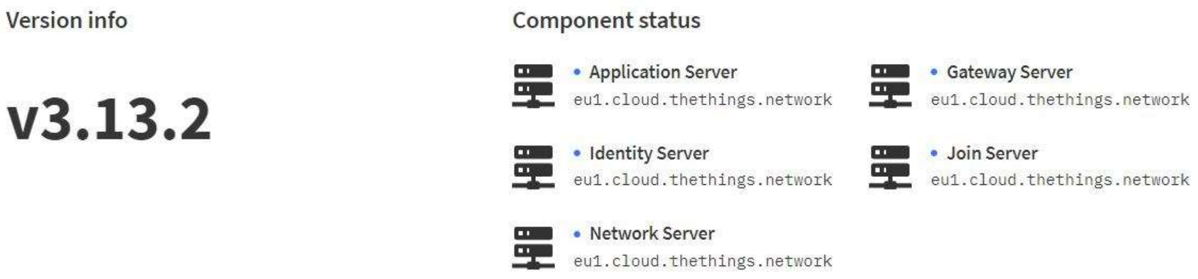
Use Seeed's Server

Server Address: seeed.thethings.industries

Uplink Port: 1700

Downlink Port: 1700

① Server Address: Please input your Server Address.
Refer to the website:



Version info

v3.13.2

Component status

- Application Server eu1.cloud.thethings.network
- Gateway Server eu1.cloud.thethings.network
- Identity Server eu1.cloud.thethings.network
- Join Server eu1.cloud.thethings.network
- Network Server eu1.cloud.thethings.network

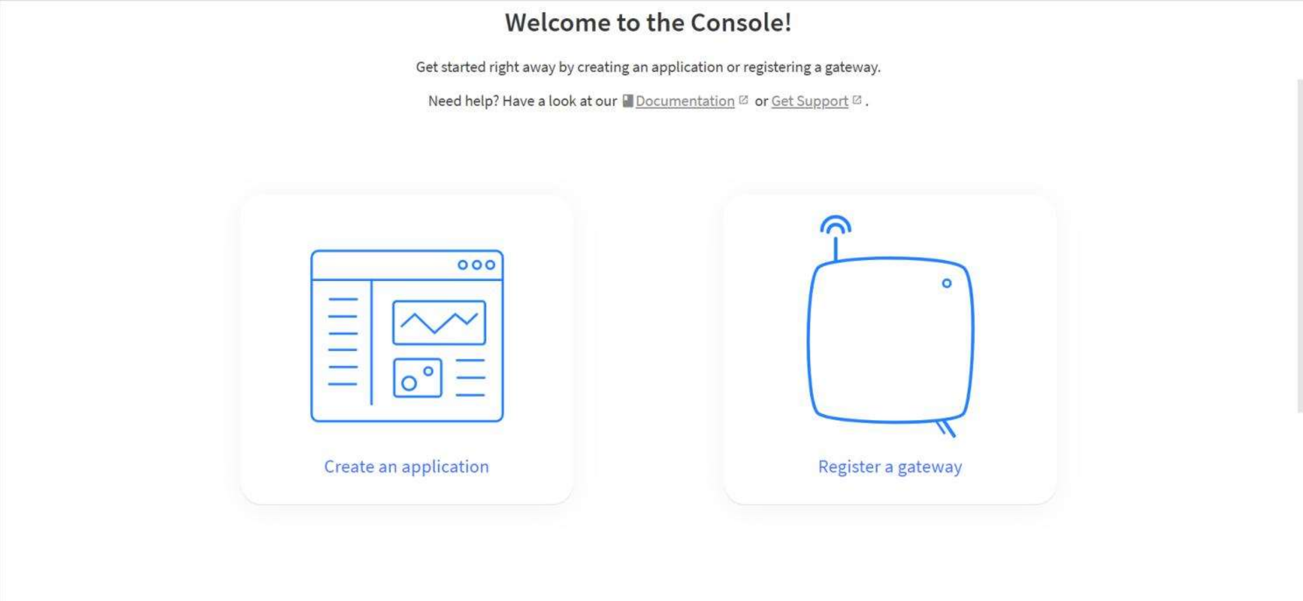
Uplink / Downlink Port (default): **1700**

(11) APPLY.

3.3 Gateway Registration on TTN

TTN website: https://www.thethingsnetwork.org
TTN console: https://console.cloud.thethings.network/
Tip: v2 will be discontinued and v3 is recommended.

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

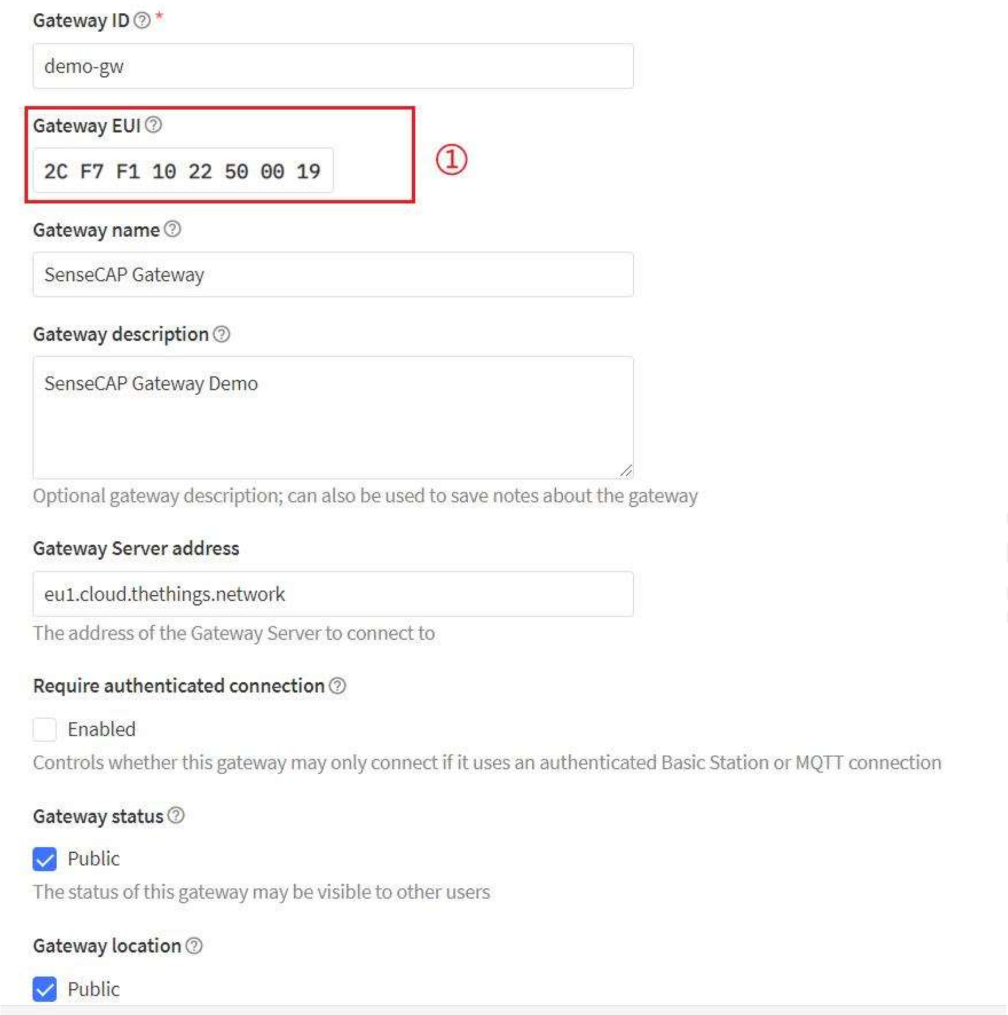


Welcome to the Console!

Get started right away by creating an application or registering a gateway.

Need help? Have a look at our Documentation or Get Support .

(2) Register Gateway



Gateway ID ⓘ *

demo-gw

Gateway EUI ⓘ

2C F7 F1 10 22 50 00 19

Gateway name ⓘ

SenseCAP Gateway

Gateway description ⓘ

SenseCAP Gateway Demo

Optional gateway description; can also be used to save notes about the gateway

Gateway Server address

eu1.cloud.thethings.network

The address of the Gateway Server to connect to

Require authenticated connection ⓘ

□ Enabled

Controls whether this gateway may only connect if it uses an authenticated Basic Station or MQTT connection

Gateway status ⓘ

☑ Public

The status of this gateway may be visible to other users

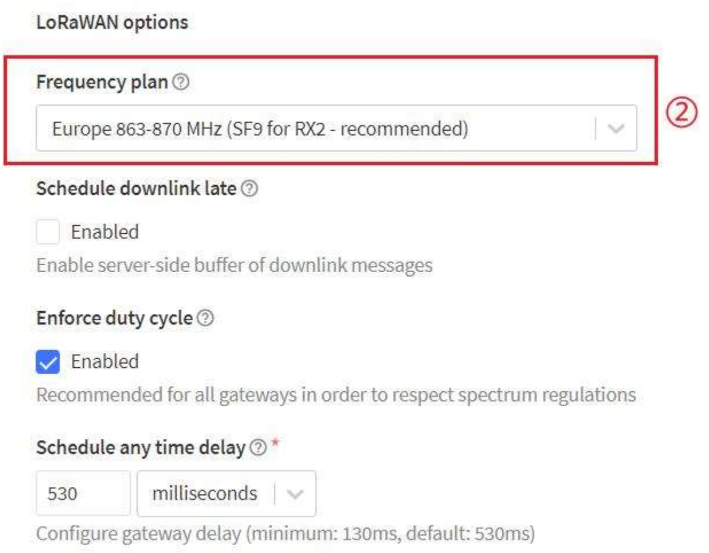
Gateway location ⓘ

☑ Public

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.

EU868	Europe 863-870 MHz (SF9 for RX2 -recommended)
US915	United States 902-928 MHz, FSB 2 (used by TTN)
AU915	Australia 915-928 MHz, FSB 2 (used by TTN)
AS923-1	Asia 920-923 MHz
AS923-2	Asia 923-925 MHz



LoRaWAN options

Frequency plan ⓘ

Europe 863-870 MHz (SF9 for RX2 - recommended)

②

Schedule downlink late ⓘ

□ Enabled

Enable server-side buffer of downlink messages

Enforce duty cycle ⓘ

☑ Enabled

Recommended for all gateways in order to respect spectrum regulations

Schedule any time delay ⓘ *

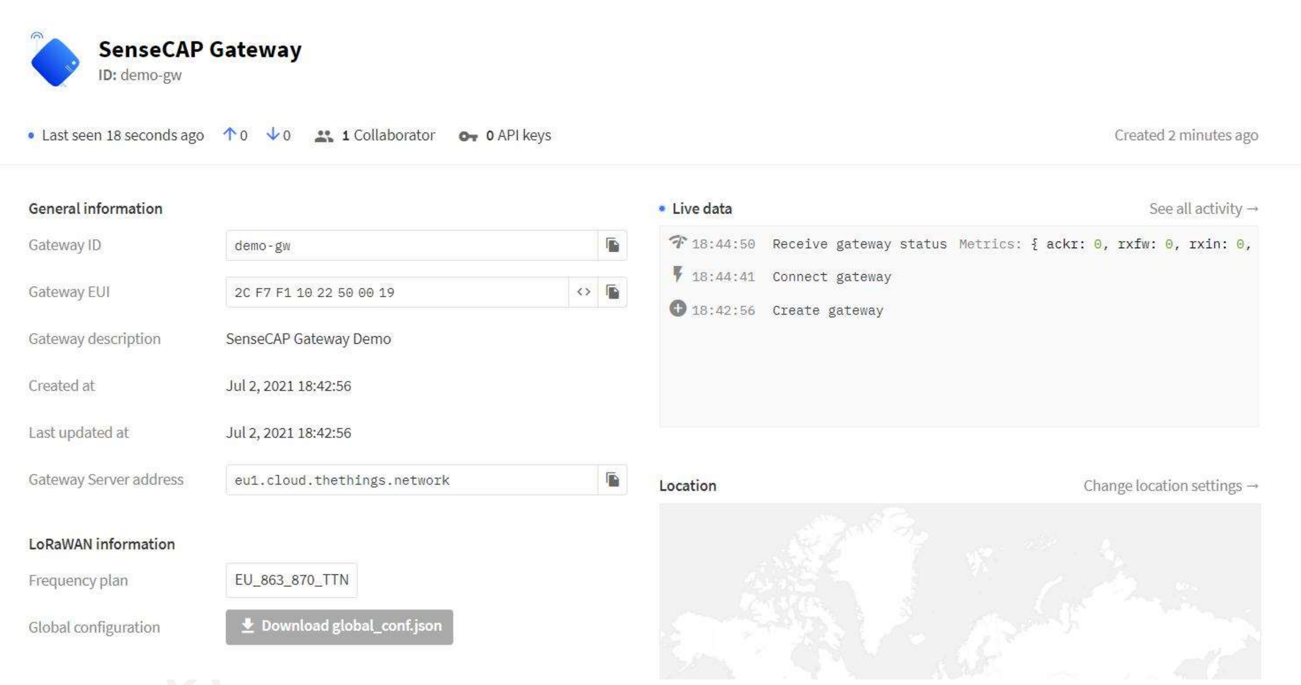
530 milliseconds

Configure gateway delay (minimum: 130ms, default: 530ms)

③ Other use default.

④ Create Gateway.

Gateway Status displays connected, indicating successful registration.



SenseCAP Gateway

ID: demo-gw

• Last seen 18 seconds ago ↑0 ↓0 1 Collaborator 0 API keys Created 2 minutes ago

General information

Gateway ID	demo-gw
Gateway EUI	2C F7 F1 10 22 50 00 19
Gateway description	SenseCAP Gateway Demo
Created at	Jul 2, 2021 18:42:56
Last updated at	Jul 2, 2021 18:42:56
Gateway Server address	eu1.cloud.thethings.network

LoRaWAN information

Frequency plan	EU_863_870_TTN
Global configuration	Download global_conf.json

• Live data See all activity →

```
18:44:50  Receive gateway status  Metrics: { ackr: 0, rxfw: 0, rxin: 0,
18:44:41  Connect gateway
18:42:56  Create gateway
```

Location Change location settings →

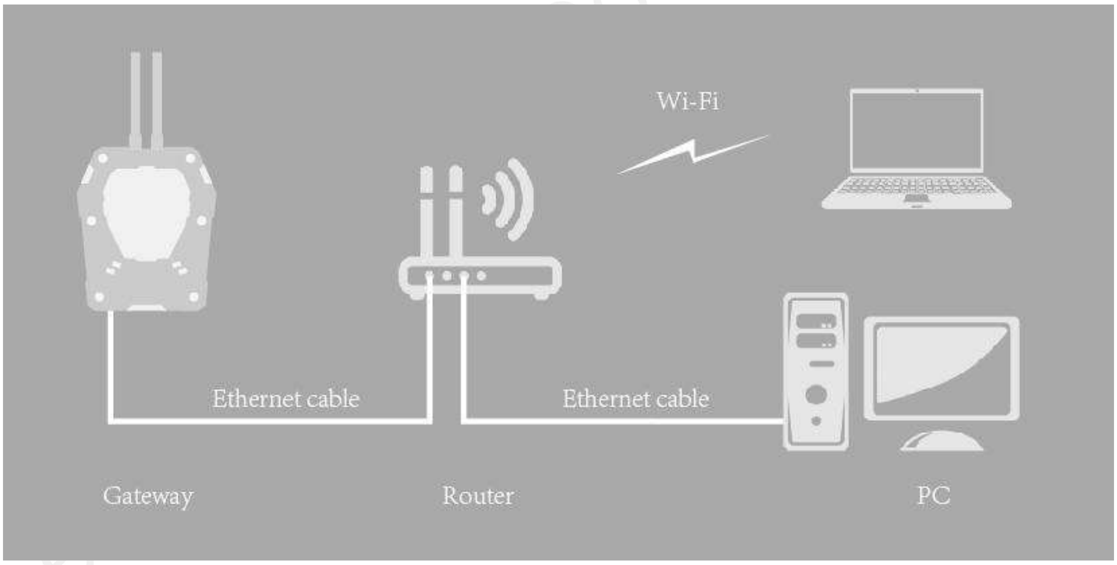
4 Add Gateway to ChirpStack LoRaWAN Network Server Stack

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.

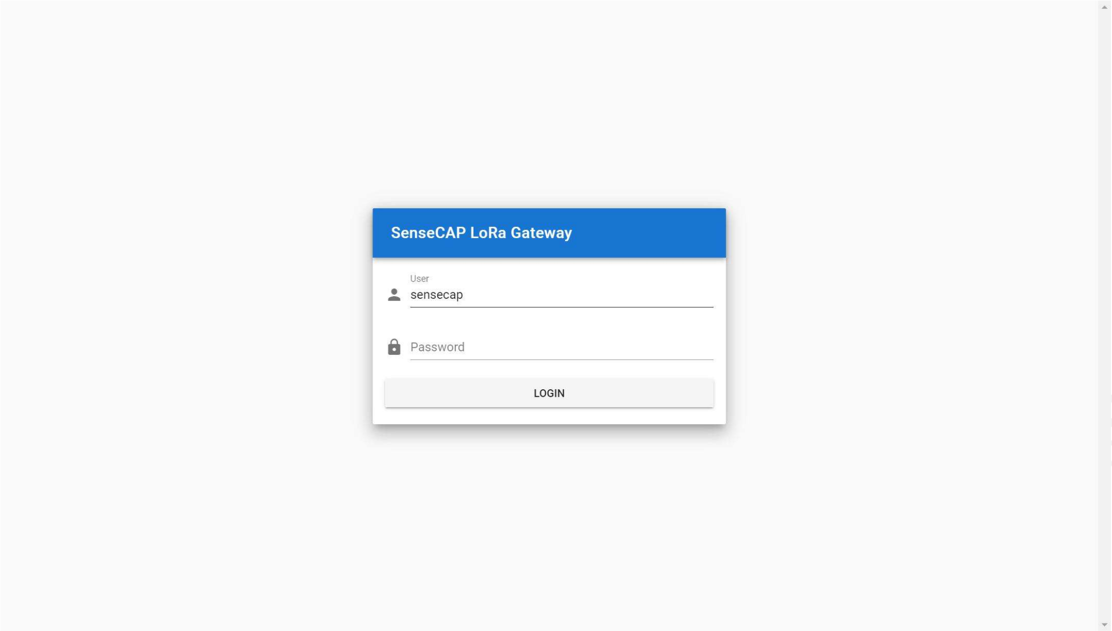
4.1 Turn on ChirpStack LoRa Server Mode

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



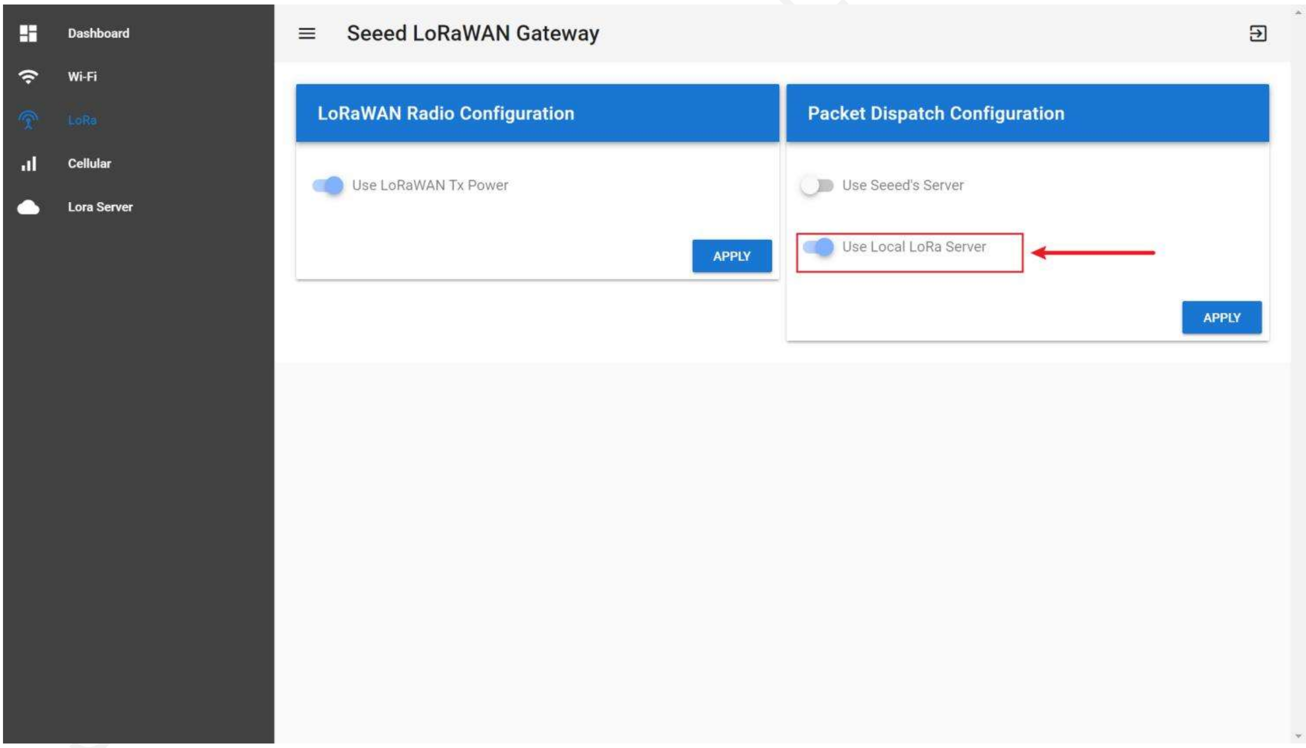
SenseCAP LoRa Gateway

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

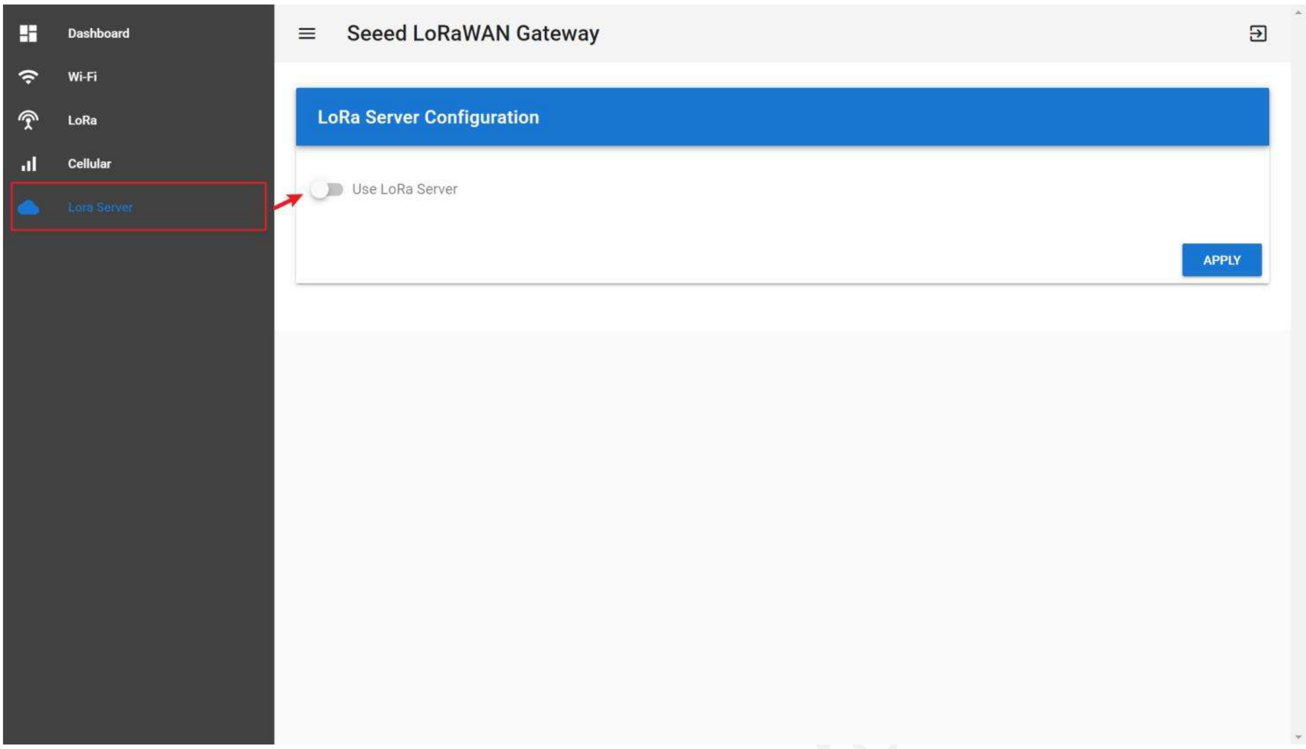
Use LoRaWAN Tx Power

Packet Dispatch Configuration

Use Seeed's Server

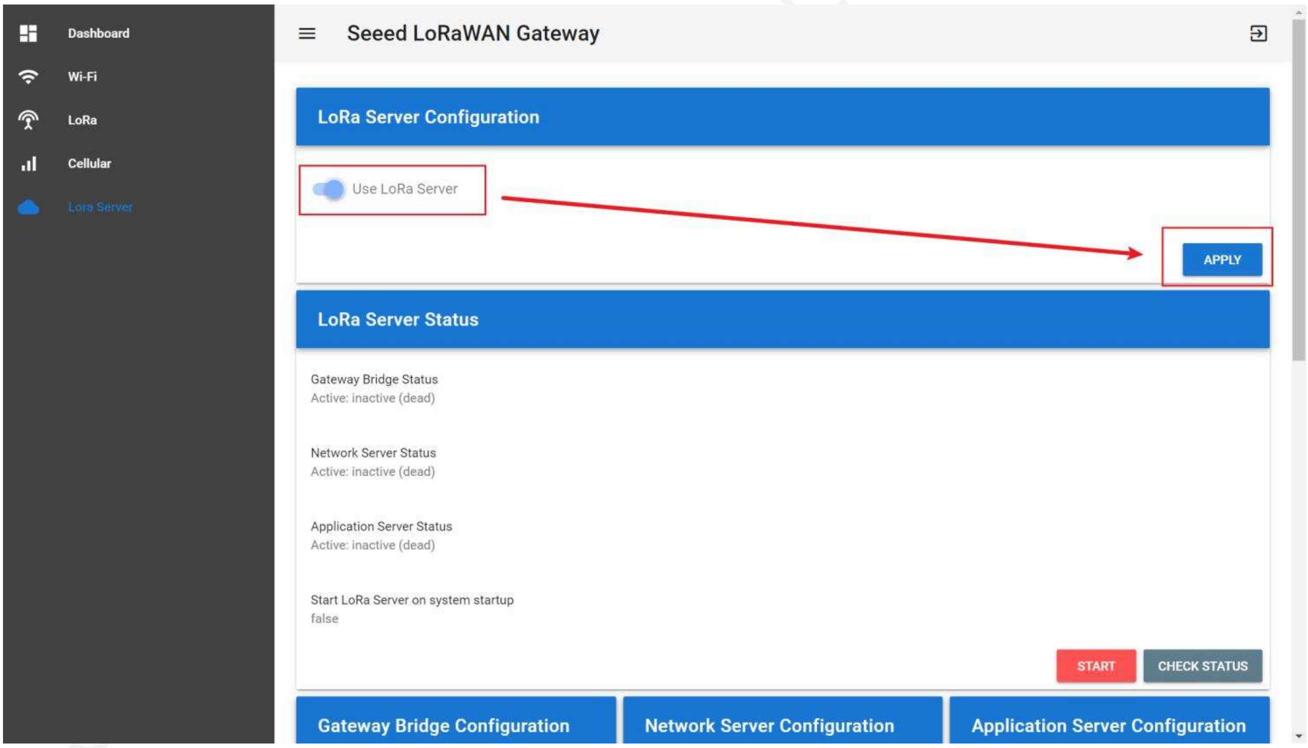
Use Local LoRa Server

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



Seeed LoRaWAN Gateway

LoRa Server Configuration



Seeed LoRaWAN Gateway

Dashboard
Wi-Fi
LoRa
Cellular
Lora Server

LoRa Server Configuration

LoRa Server Status

Gateway Bridge Status
Active: inactive (dead)

Network Server Status
Active: inactive (dead)

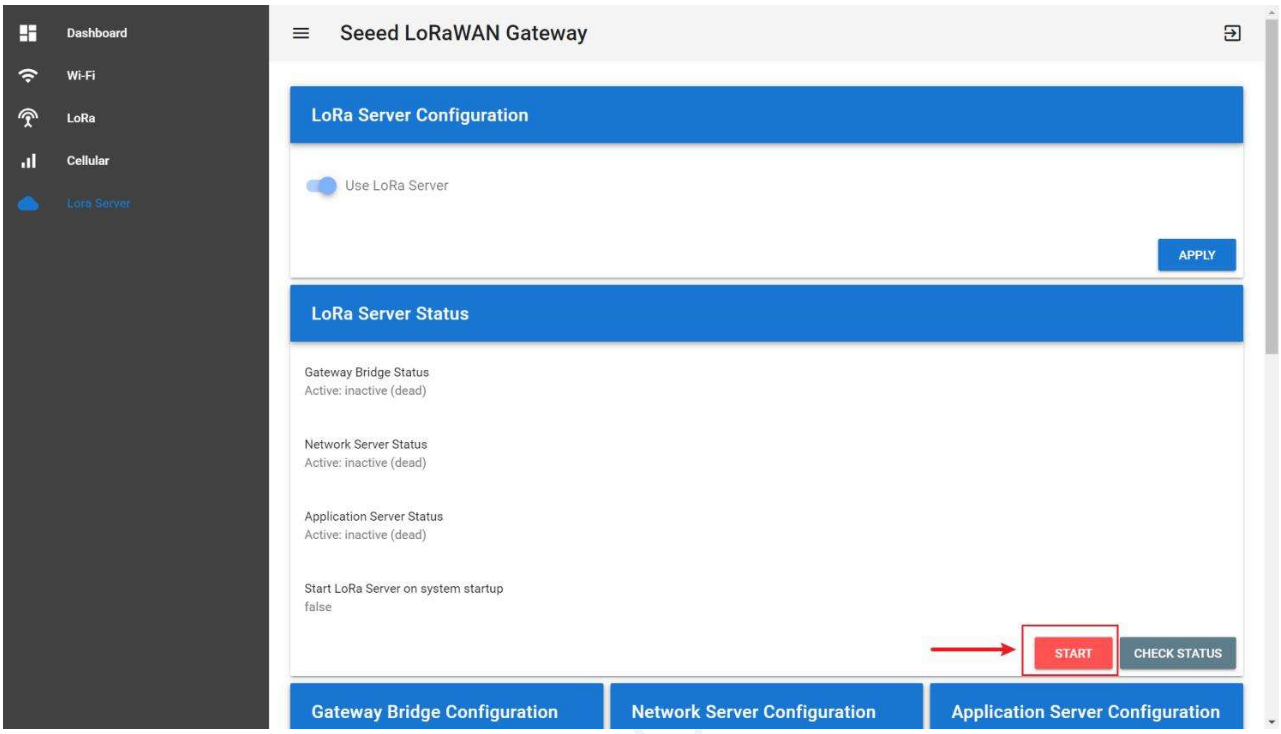
Application Server Status
Active: inactive (dead)

Start LoRa Server on system startup
false

START
CHECK STATUS
Gateway Bridge Configuration
Network Server Configuration
Application Server Configuration

4.2 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)

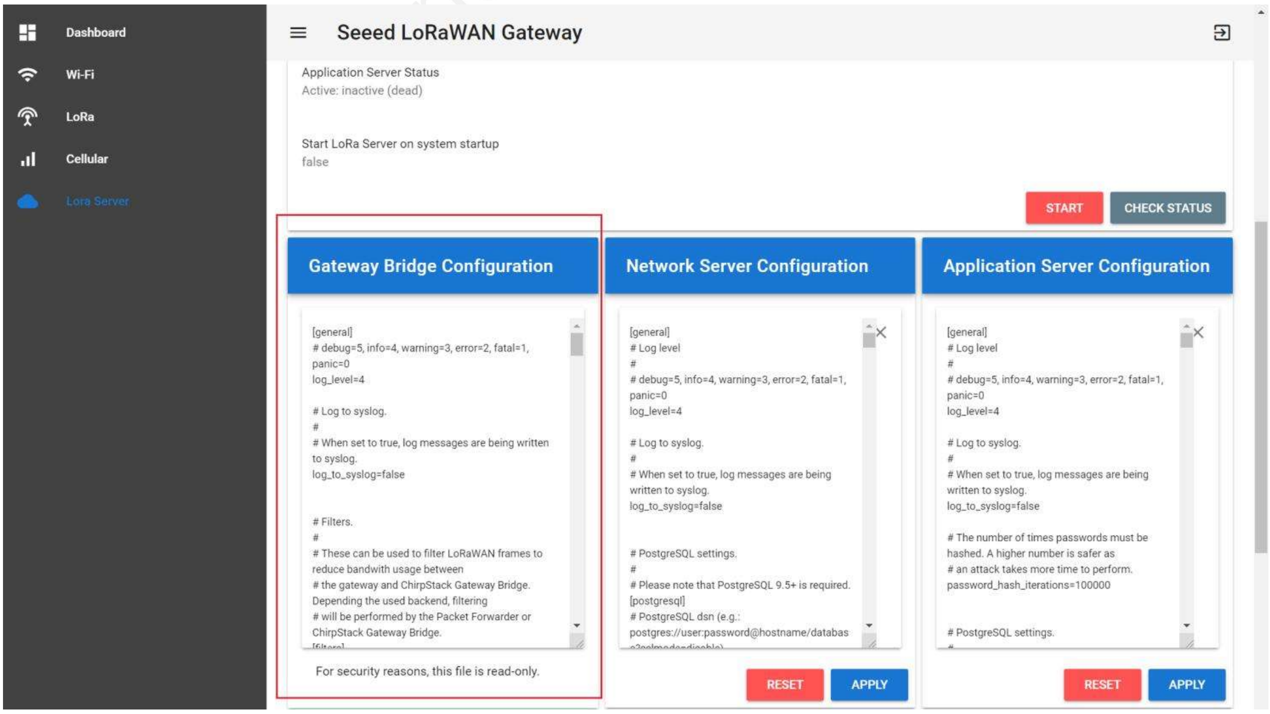
Start LoRa Server on system startup
false

(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.
```

For security reasons, this file is read-only.

Network Server Configuration

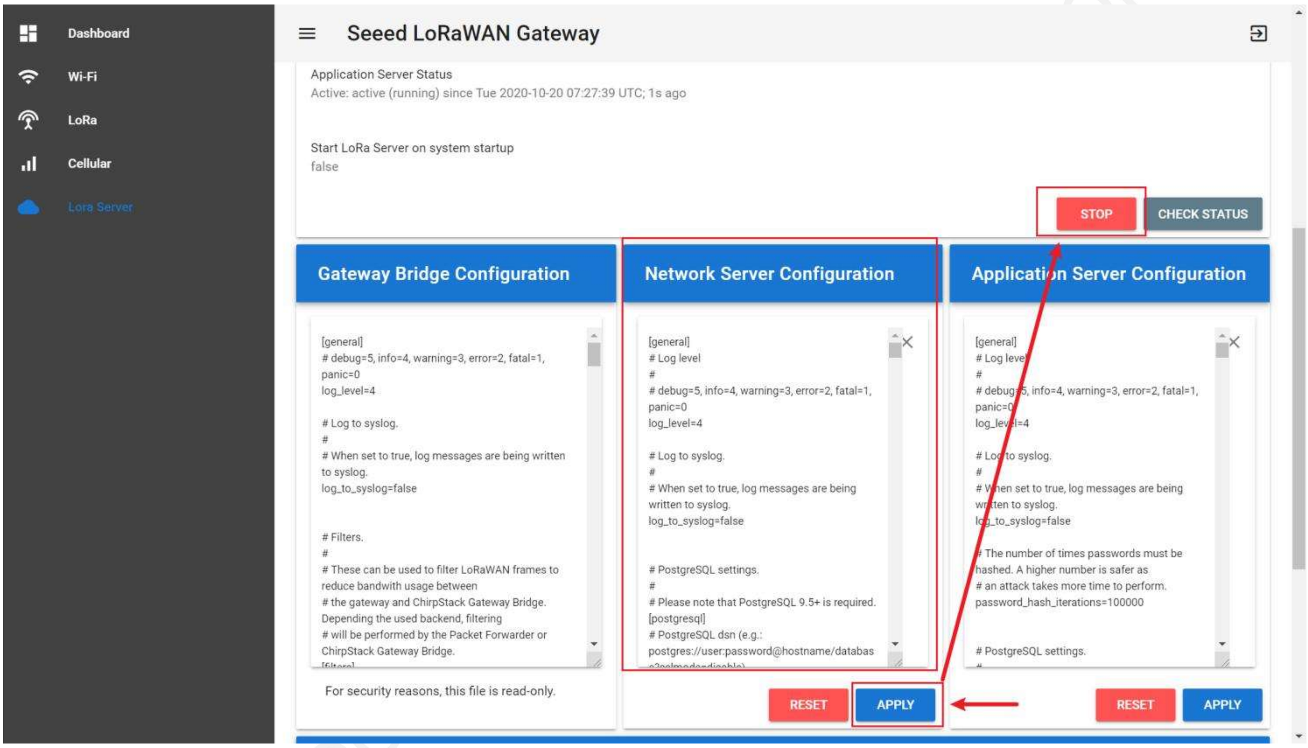
```
[general]
# Log level
#
# debug=5, info=4, warning=3, error=2, fatal=1,
panic=0
log_level=4

# 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.
```

(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

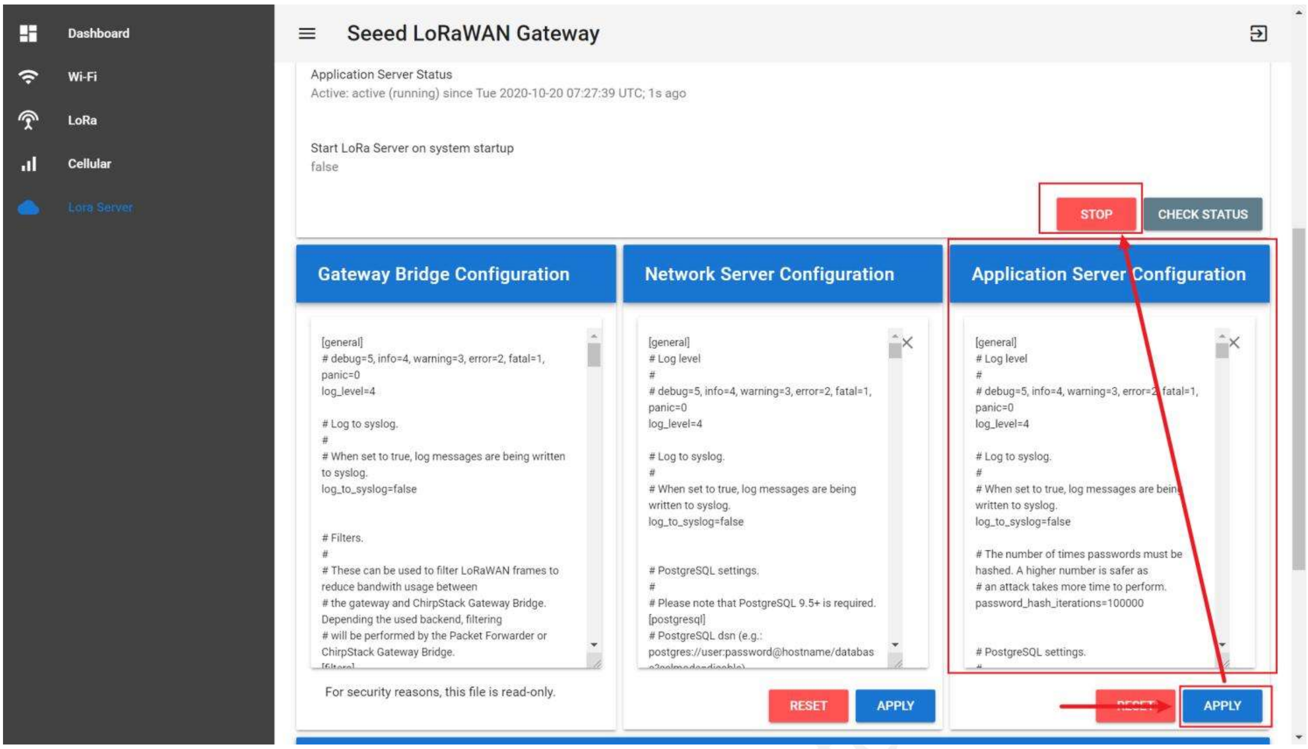
```
[general]
# Log level
#
# debug=5, info=4, warning=3, error=2, fatal=1, panic=0
log_level=4

# 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
```

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.
[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.
#
```

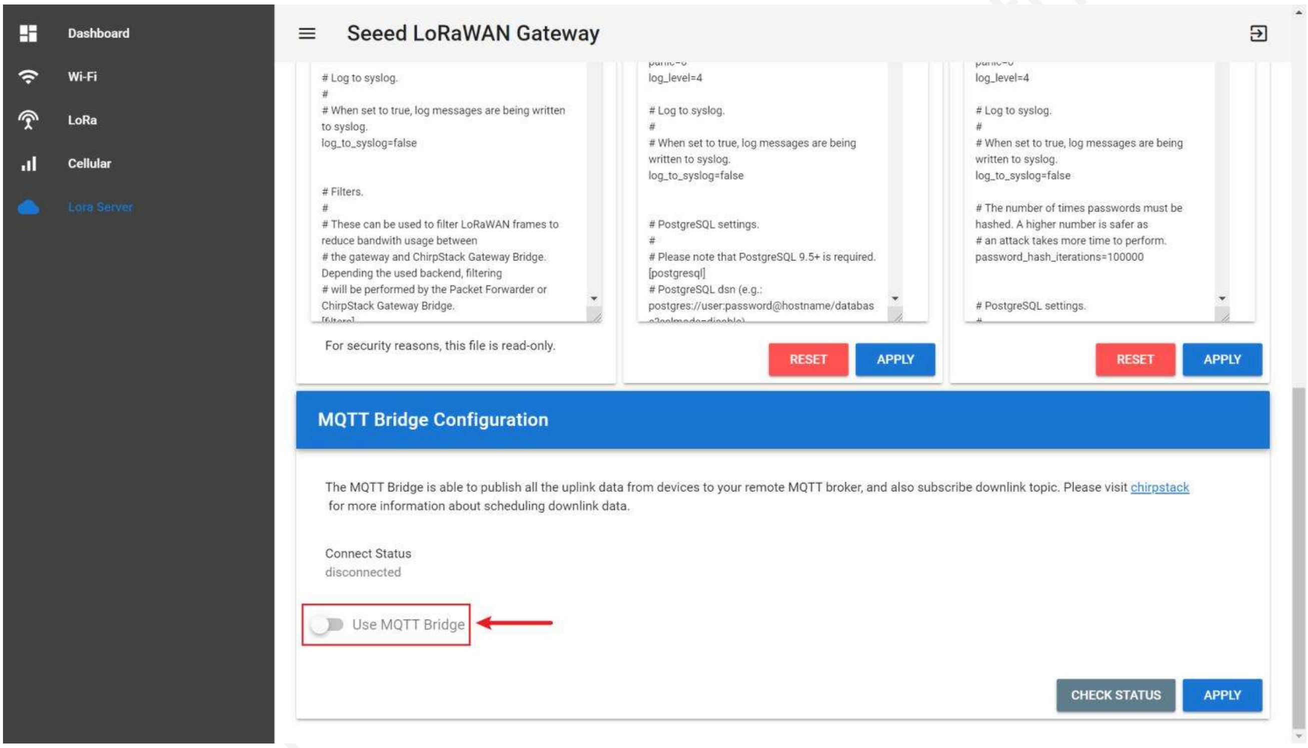
(4) If you have the wrong configuration, click "RESET" to restore the default configuration.

4.3 **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.

4.3.1 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.
```

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

(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

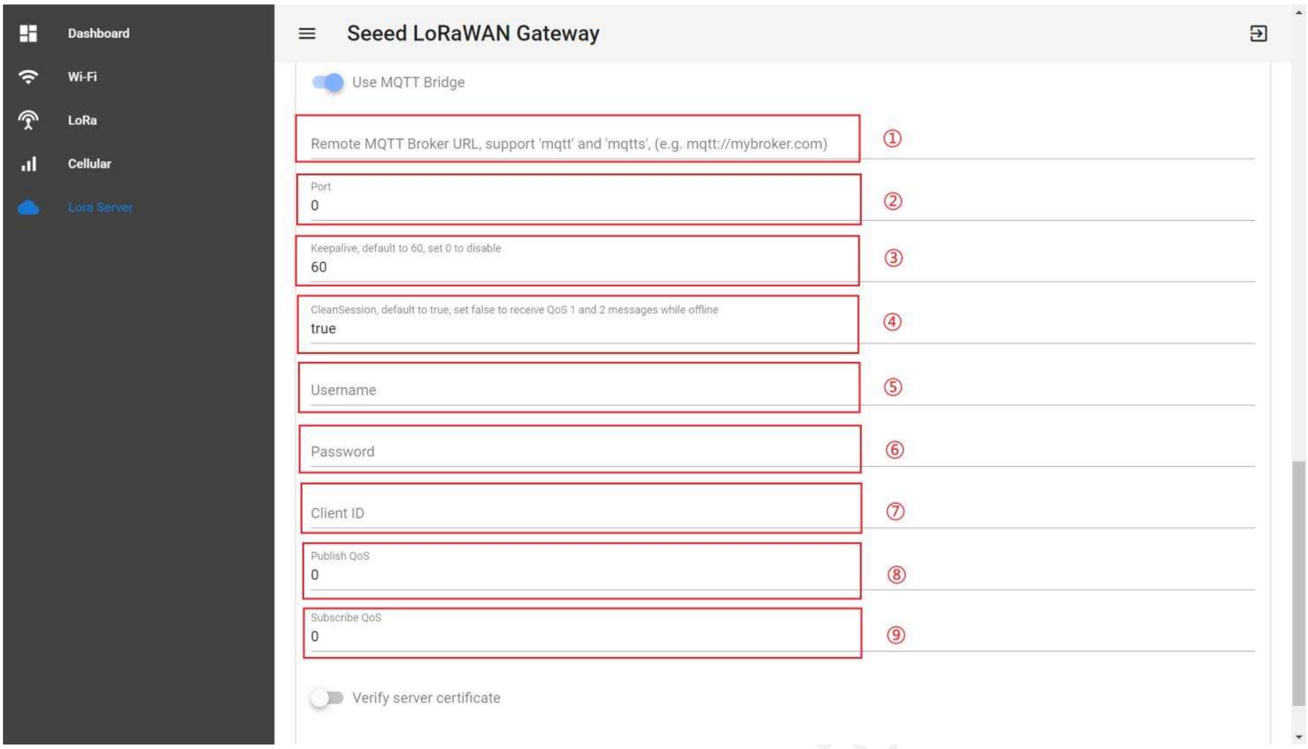
Client ID: Custom the name, and each Client ID is unique to the same MQTT server.

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

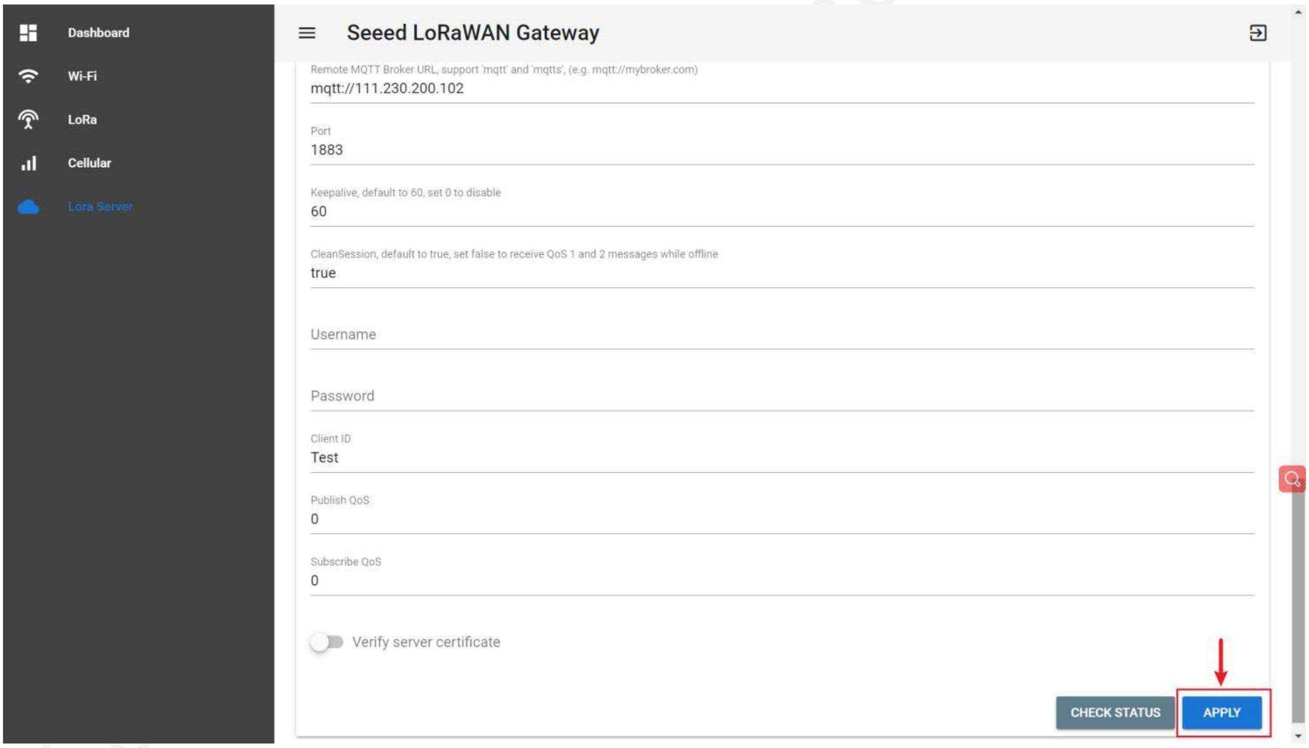
Password

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

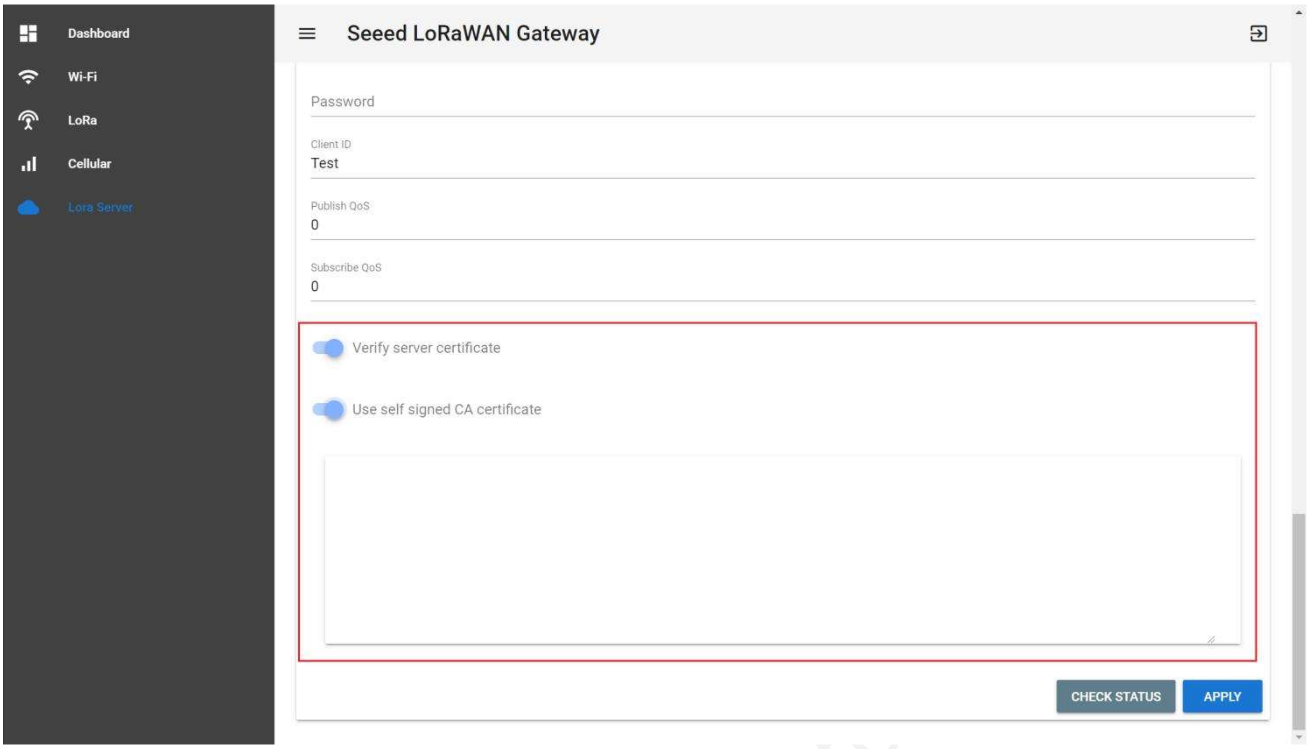
Client ID
Test

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

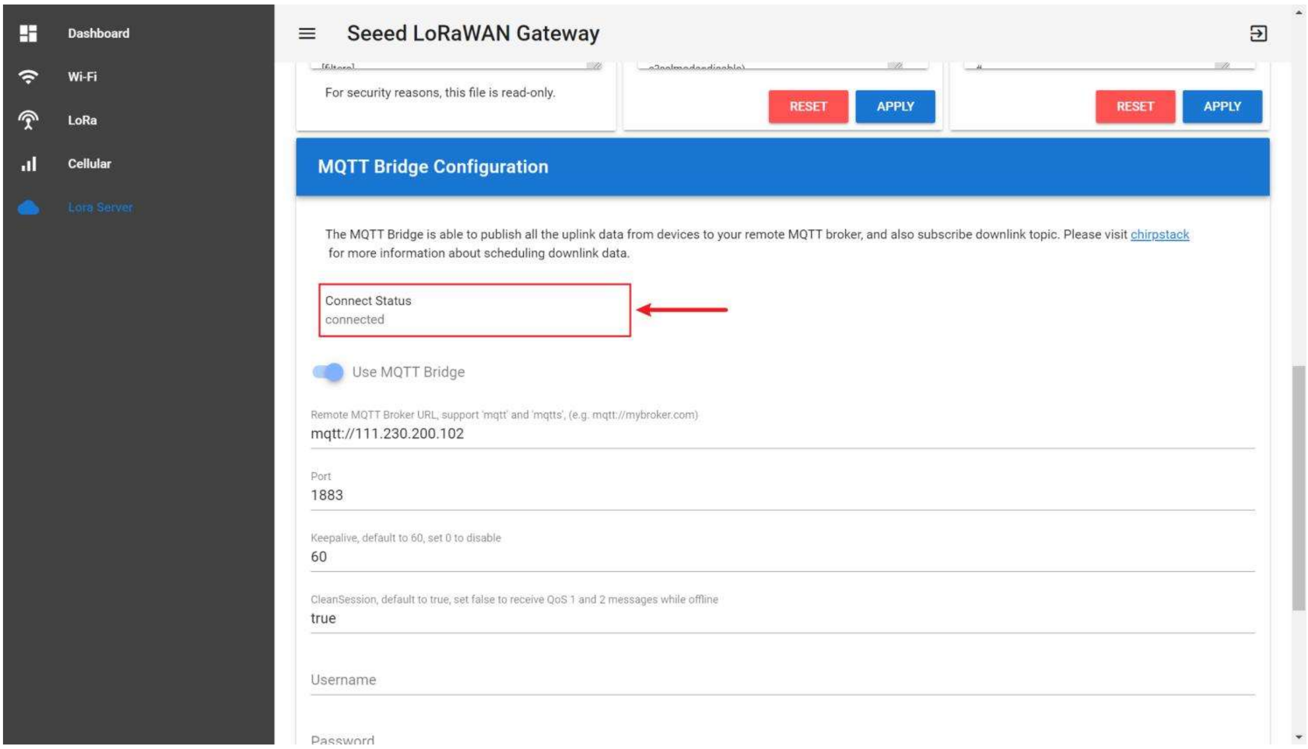
Publish QoS
0

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.

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
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

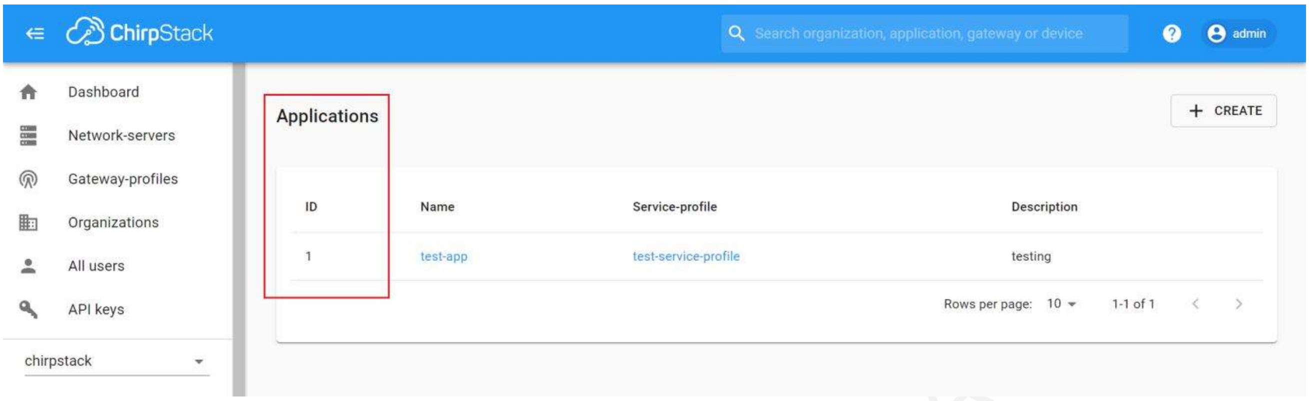
Username

Password

4.3.2 MQTT Client Configuration

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

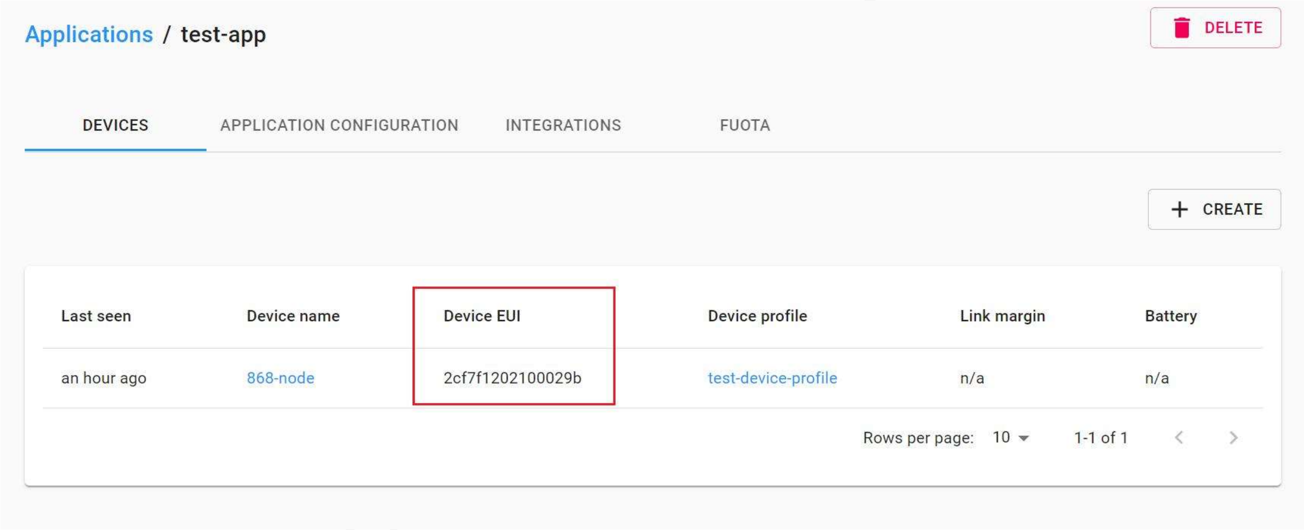
ApplicationID: the Application ID.



Applications

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

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

(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
```

4.3.3 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.

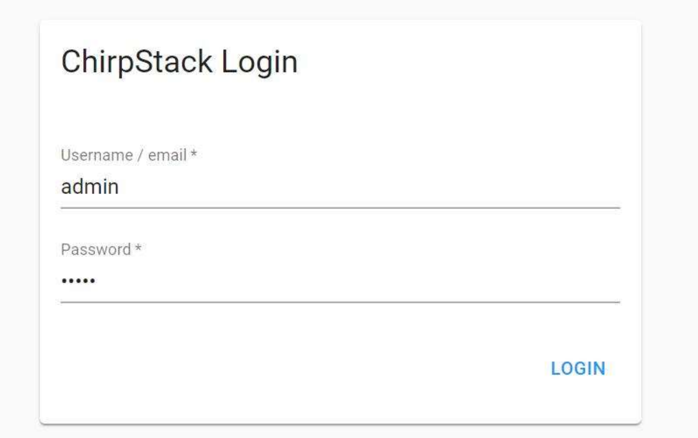
Example payload:

```
{
    "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}
    }
}
```

4.4 ChirpStack Application Server

4.4.1 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

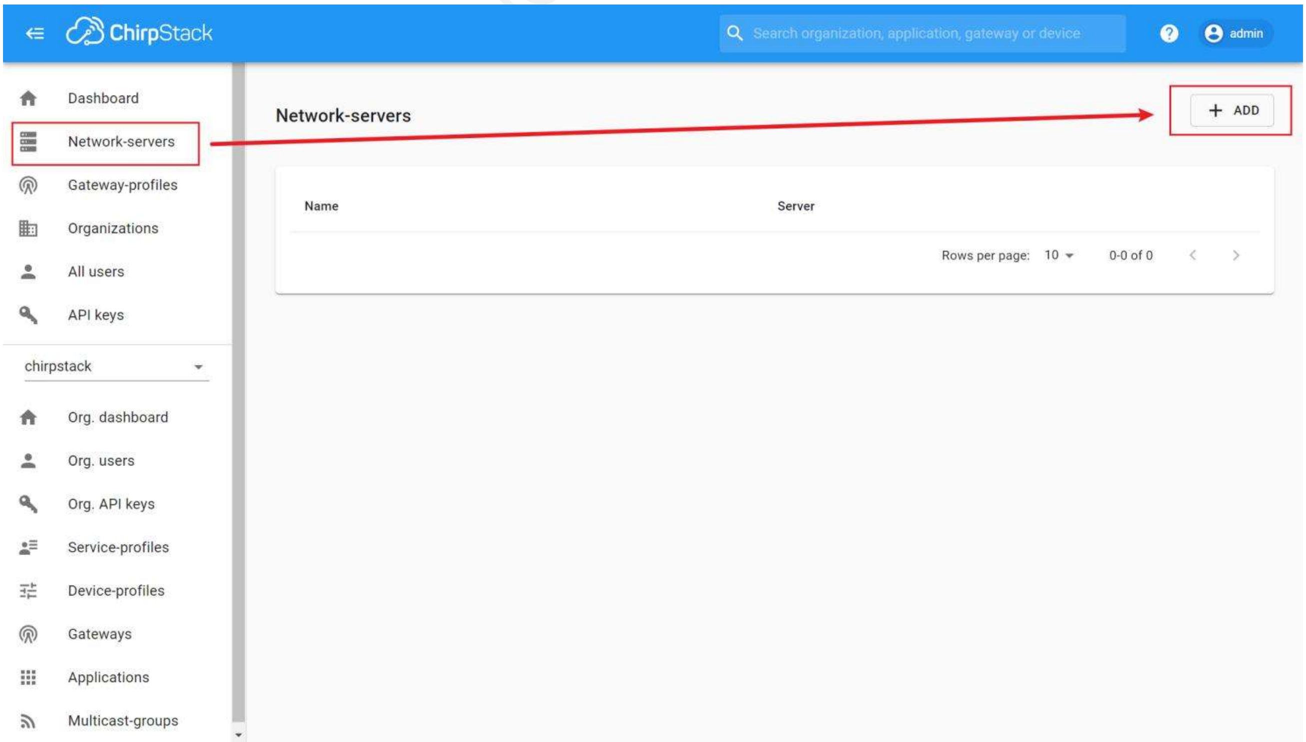


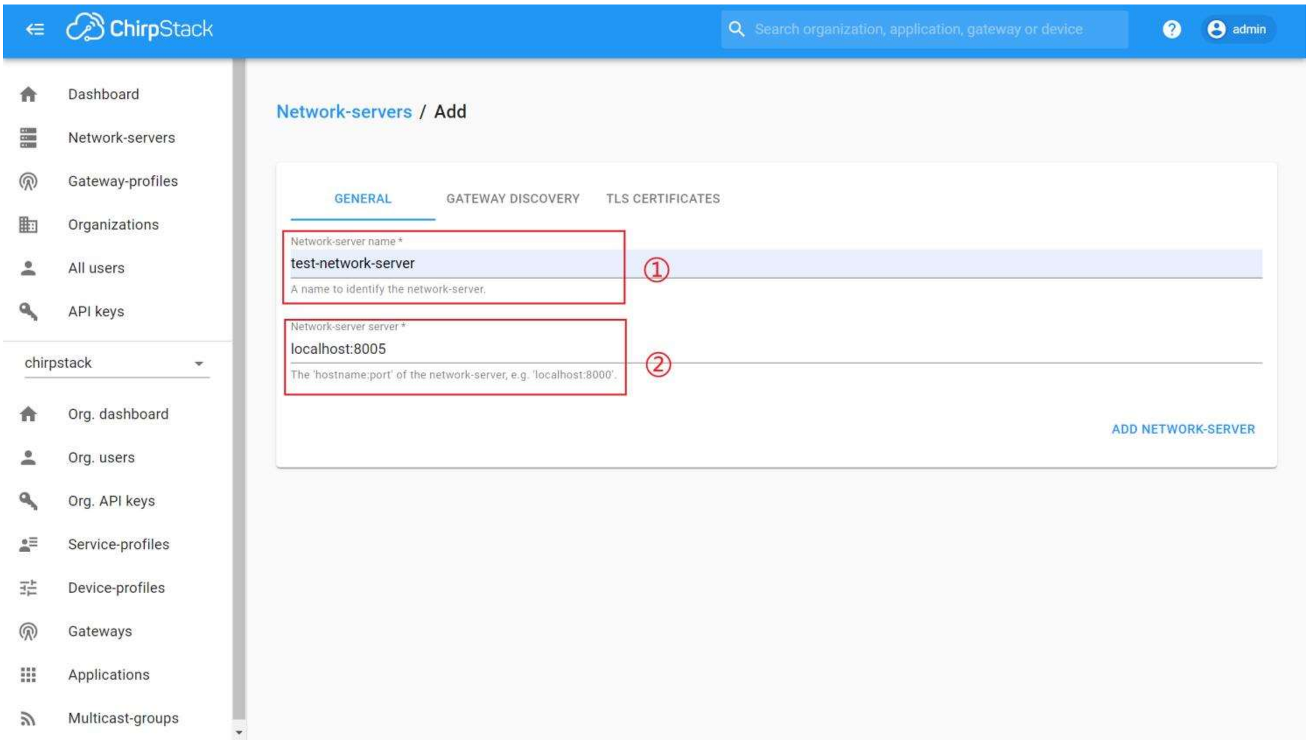
ChirpStack Login

Username / email *
admin

Password *
•••••

4.4.2 Add the Network-servers





Network-servers / Add

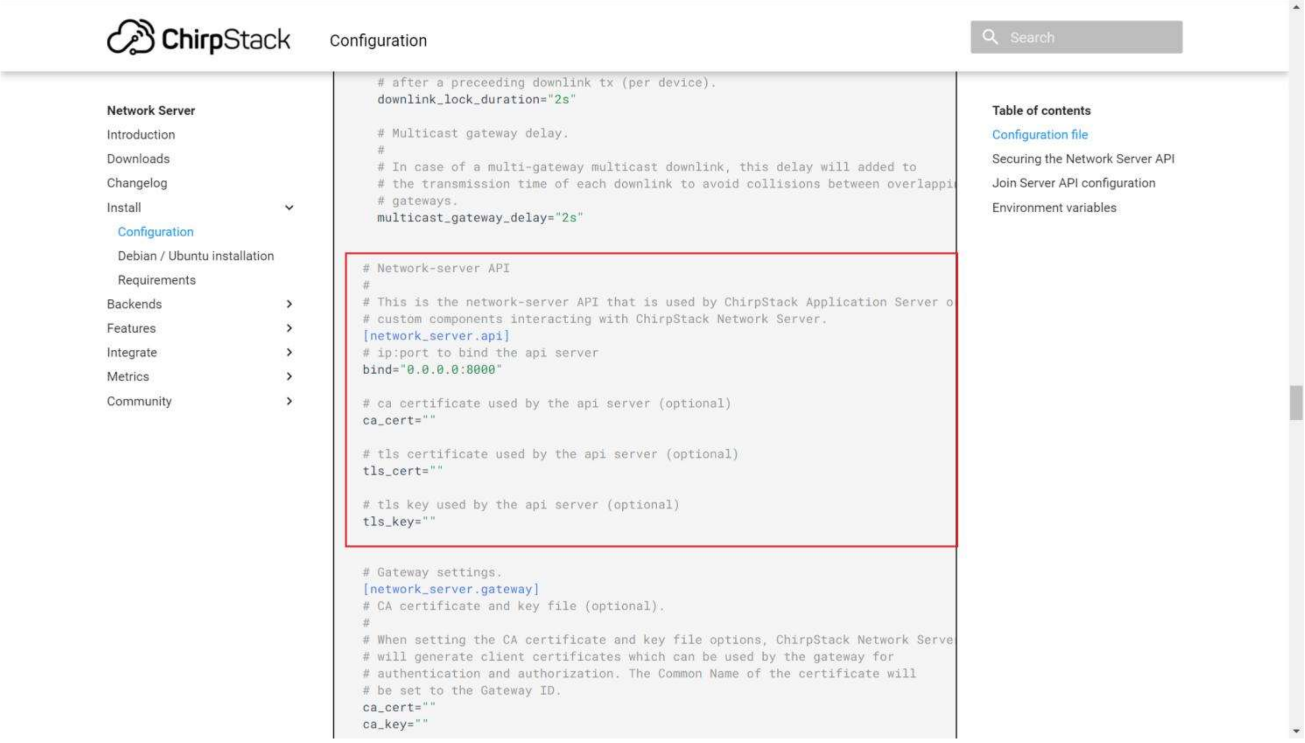
Network-server name *
test-network-server
A name to identify the network-server.

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

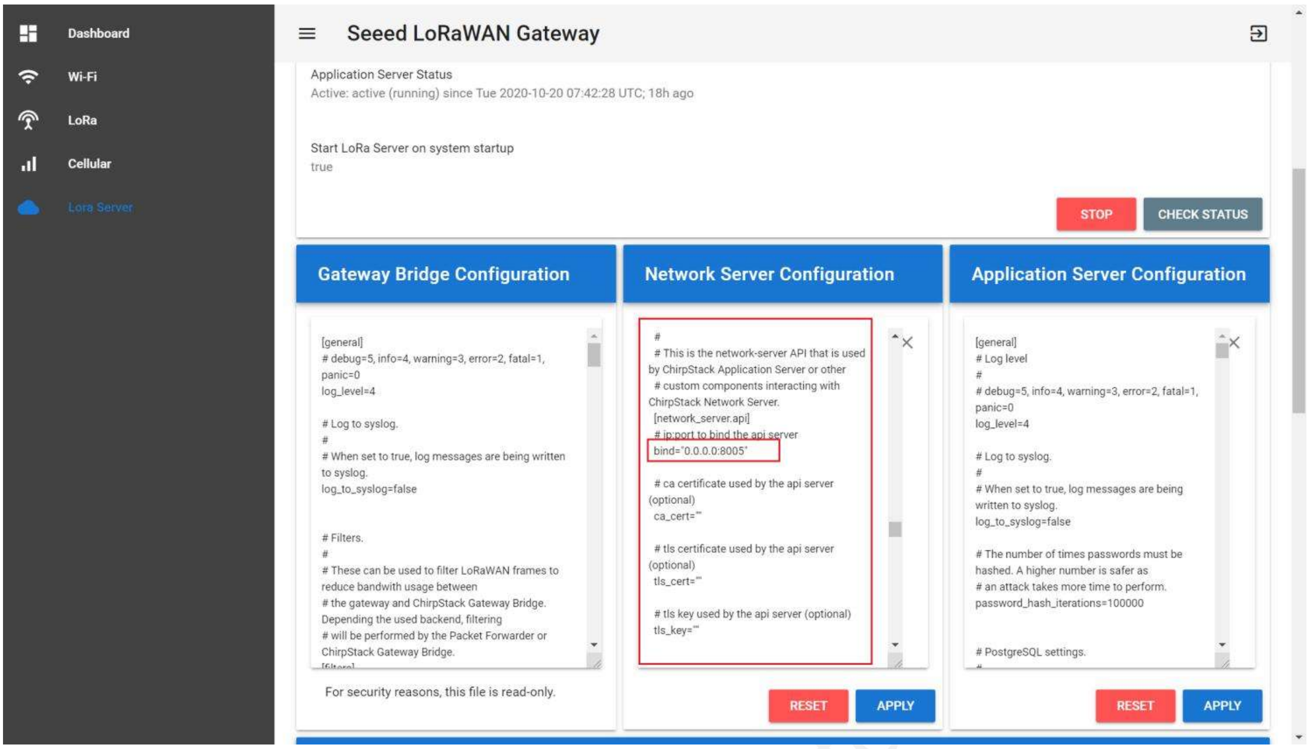
① Network-server name: custom name.

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

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.

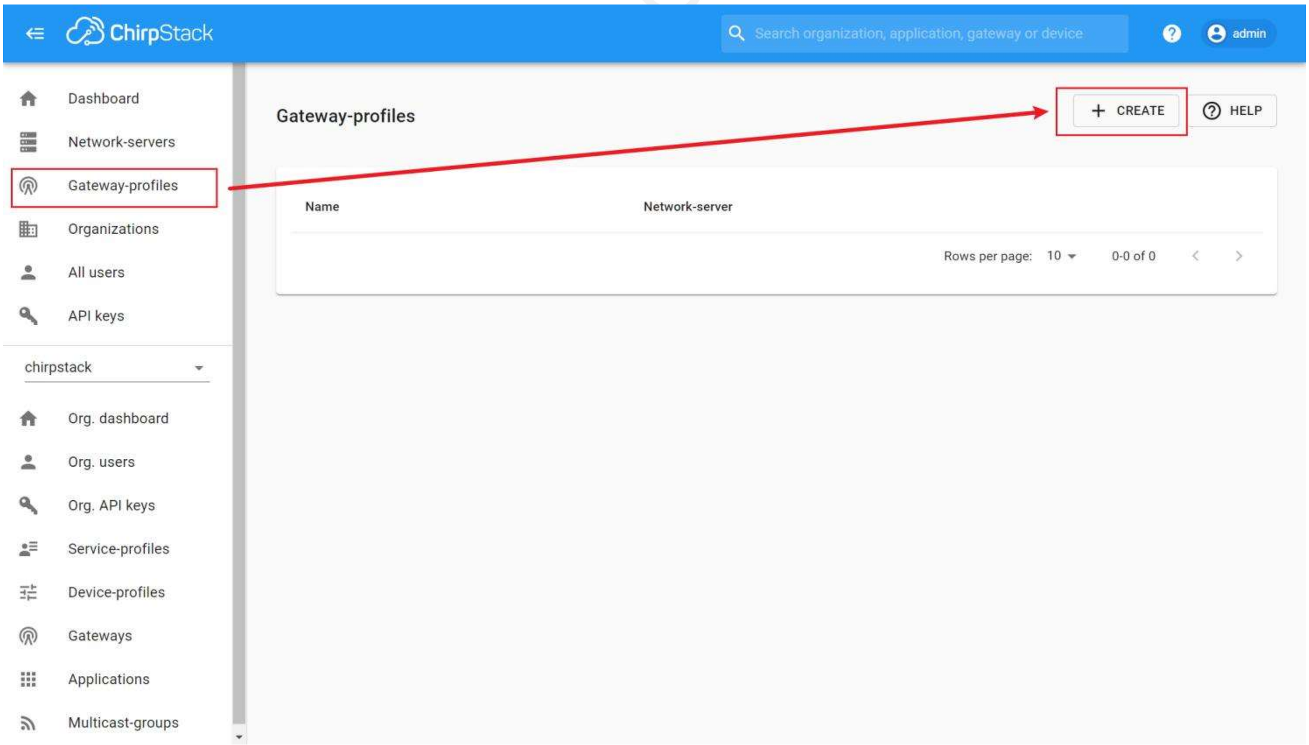
Network Server Configuration

```
 #
 # 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=""
```

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.
#
```

4.4.3 Create the Gateway-profiles



Name	Network-server

Rows per page: 10 0-0 of 0

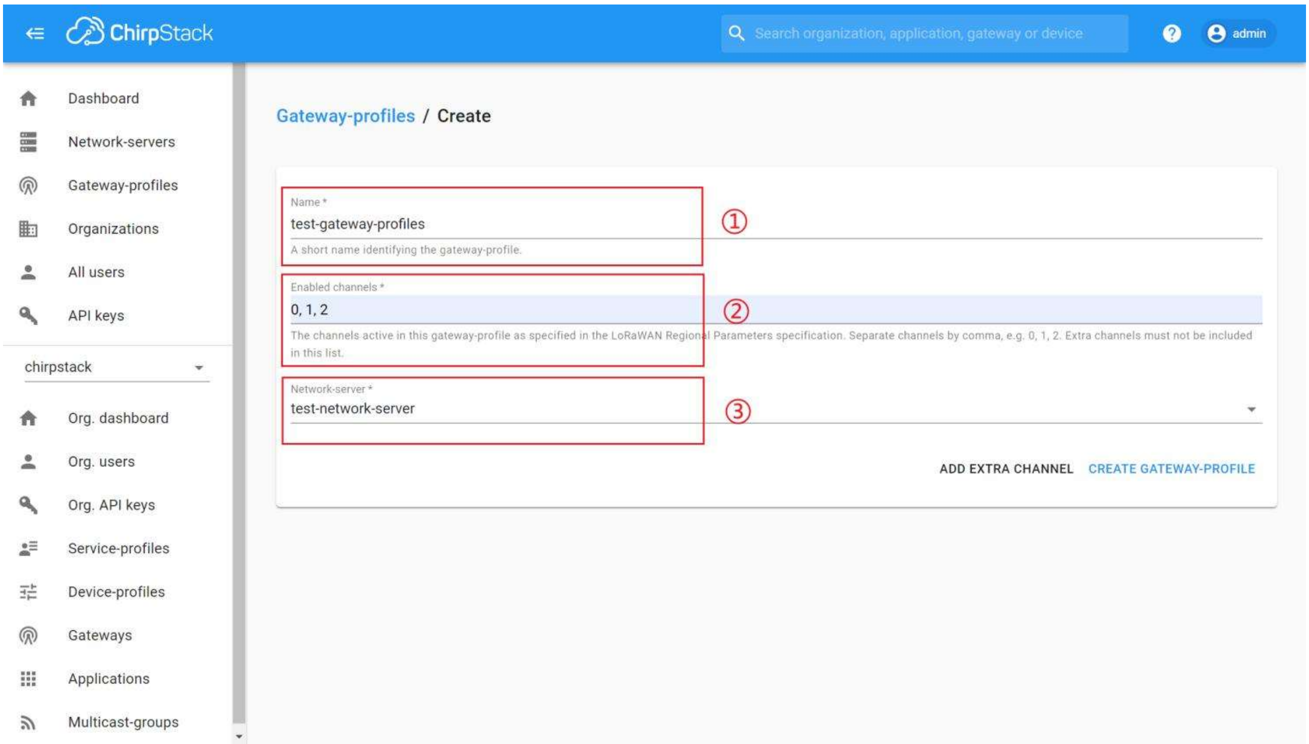
① Name: custom name.

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.

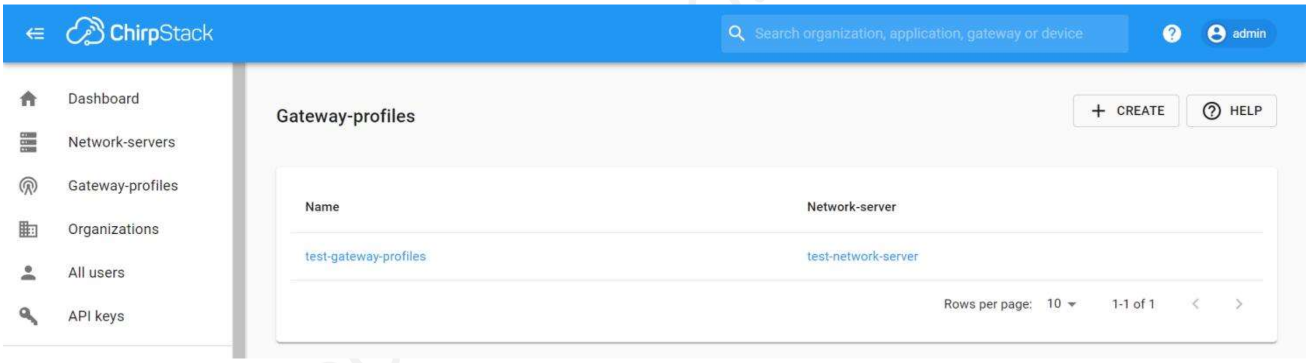


Name *
test-gateway-profiles
A short name identifying the gateway-profile.

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

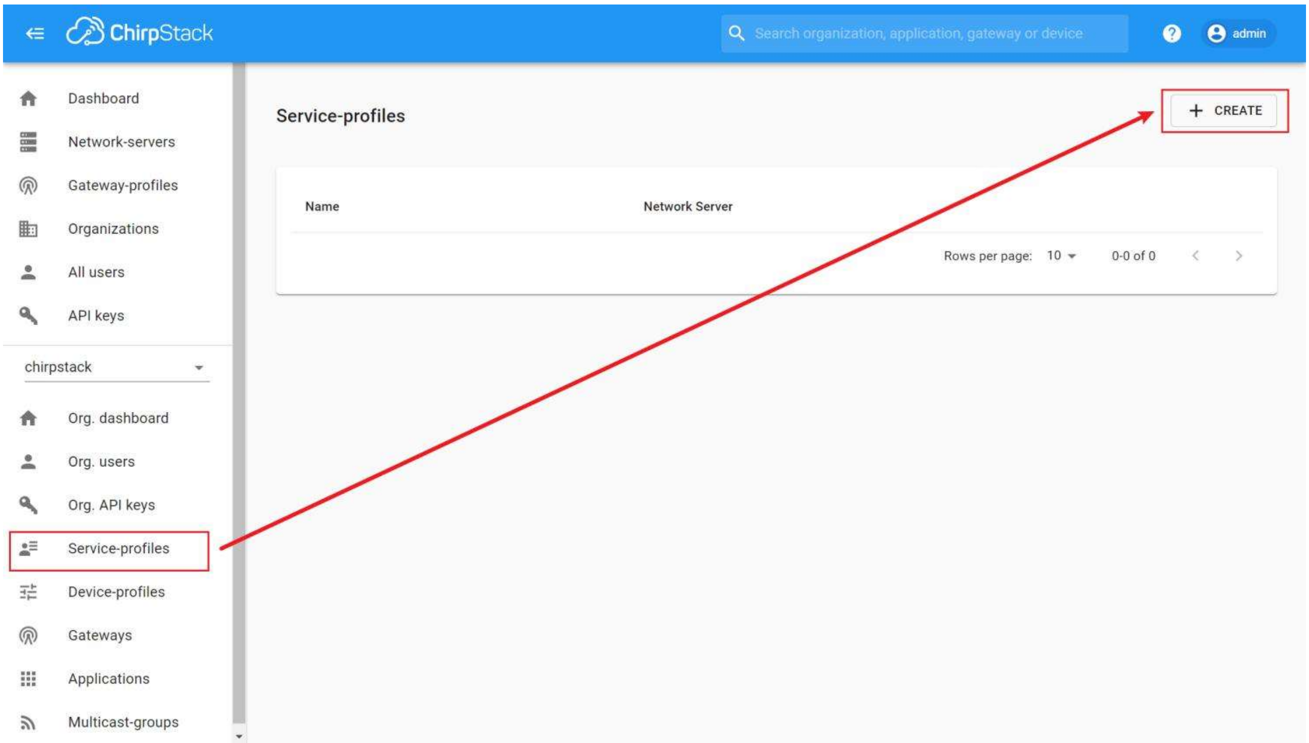
Click the "GREATE GATEWAY-PROFILE".

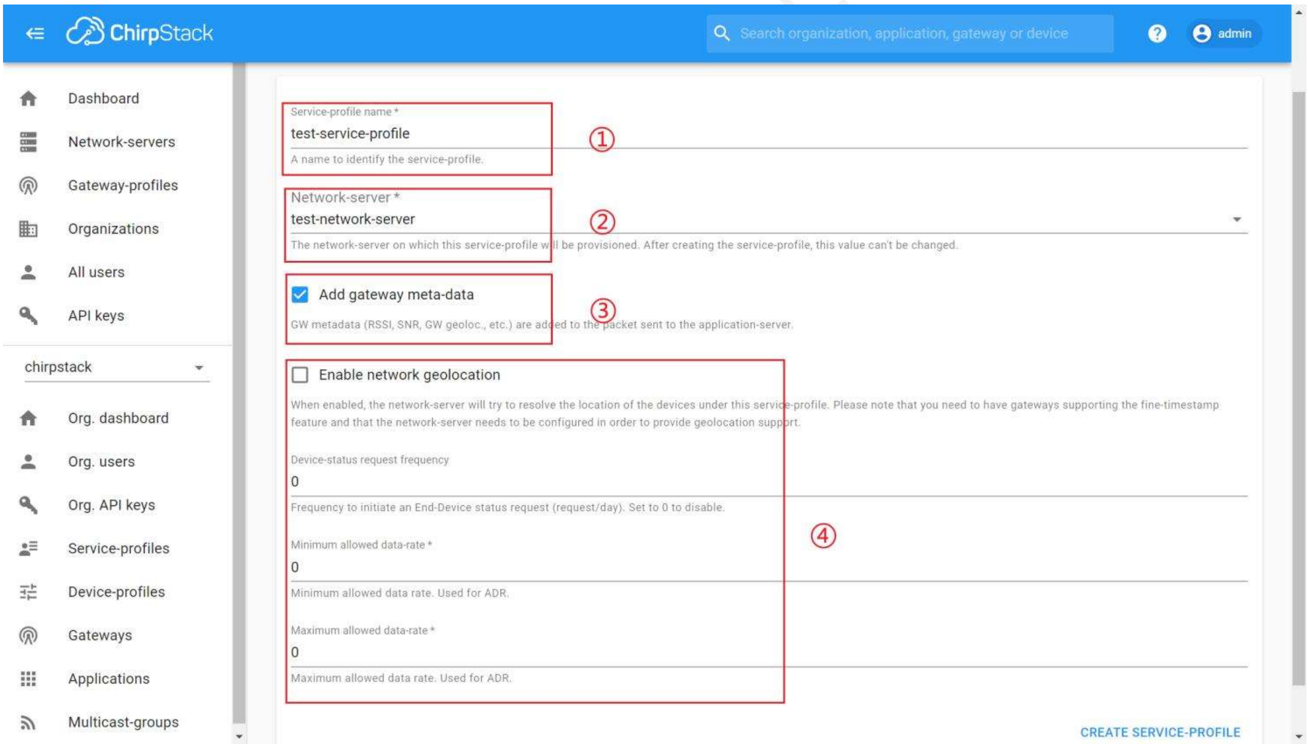


Name	Network-server
test-gateway-profiles	test-network-server

Rows per page: 10 1-1 of 1

4.4.4 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.

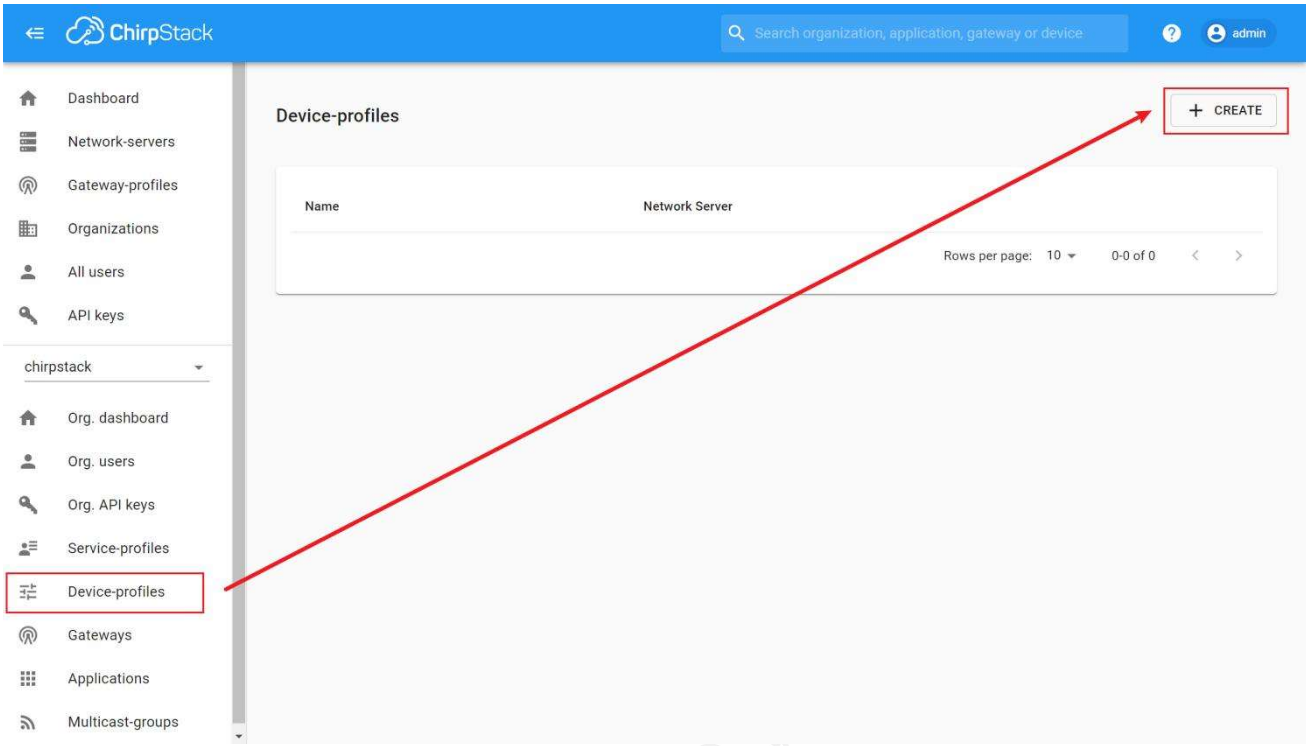
Device-status request frequency
0
Frequency to initiate an End-Device status request (request/day). Set to 0 to disable.

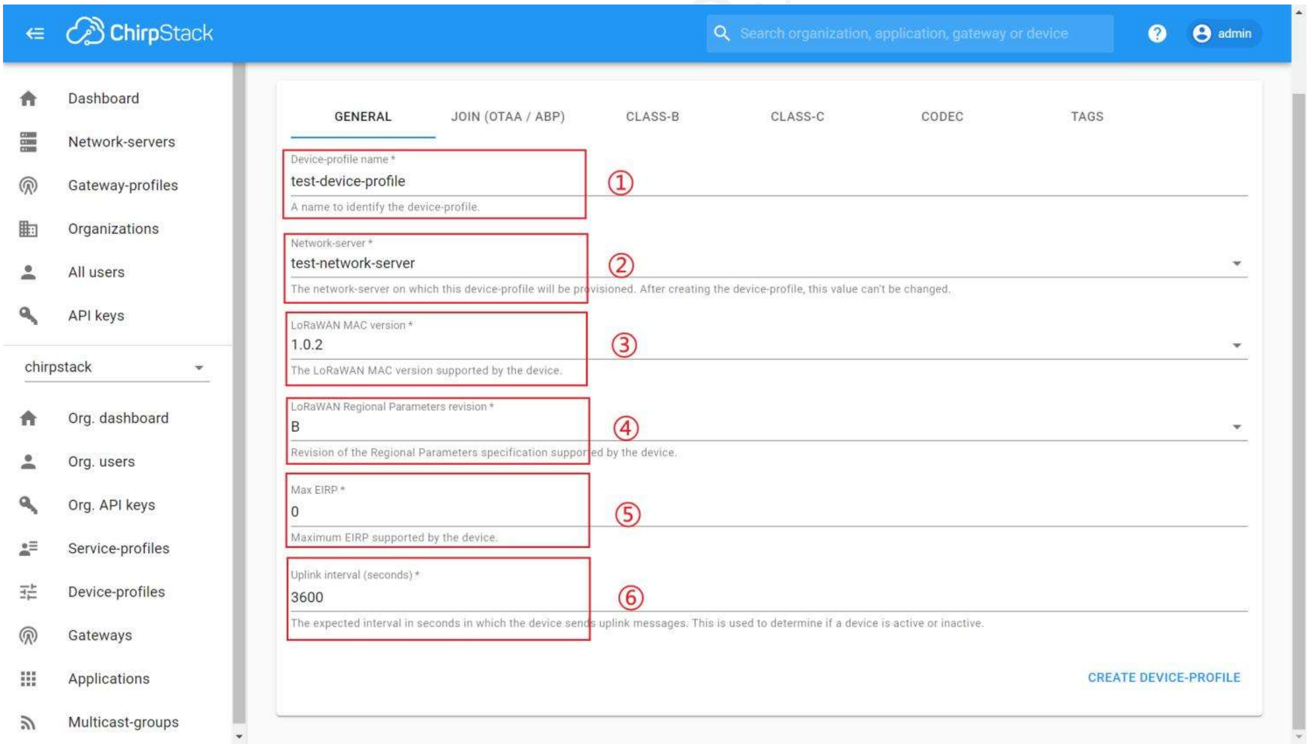
Minimum allowed data-rate *
0
Minimum allowed data rate. Used for ADR.

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.

4.4.5 Create the Device-profiles





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

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

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

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

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

⑥ Uplink interval (seconds) *: 3600
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.

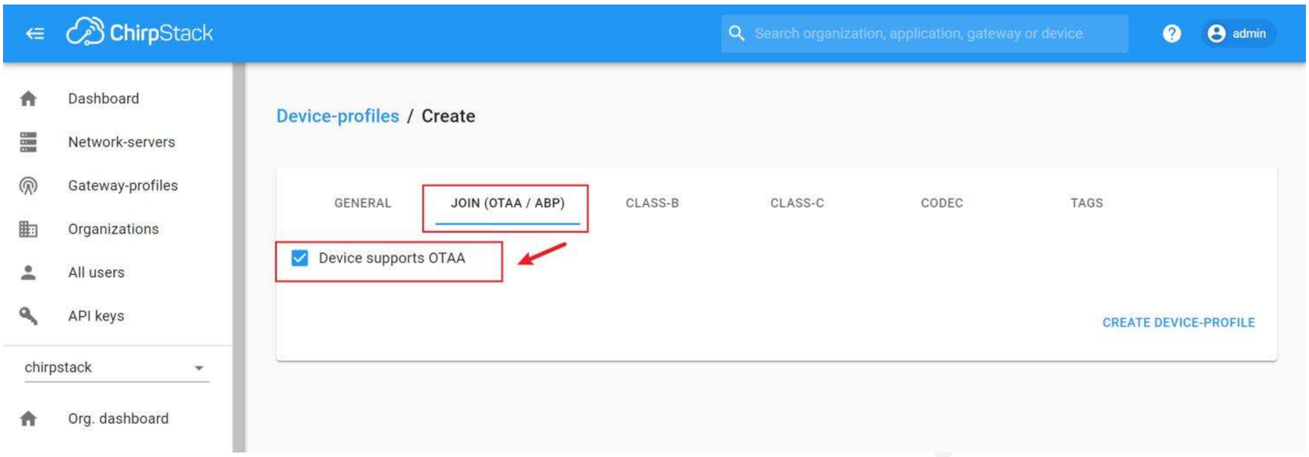
③ LoRaWAN MAC version: 1.0.2 (only for SenseCAP Node)

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”.



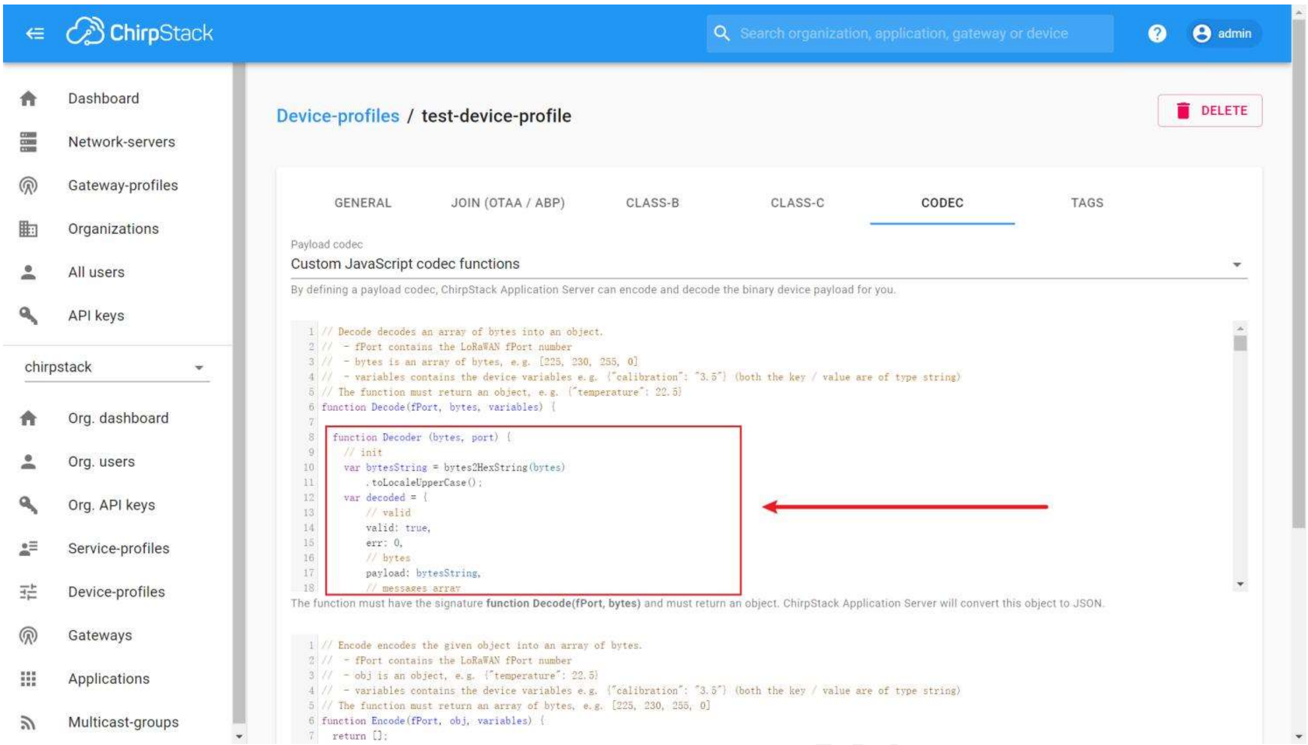
☑ Device supports OTAA

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.

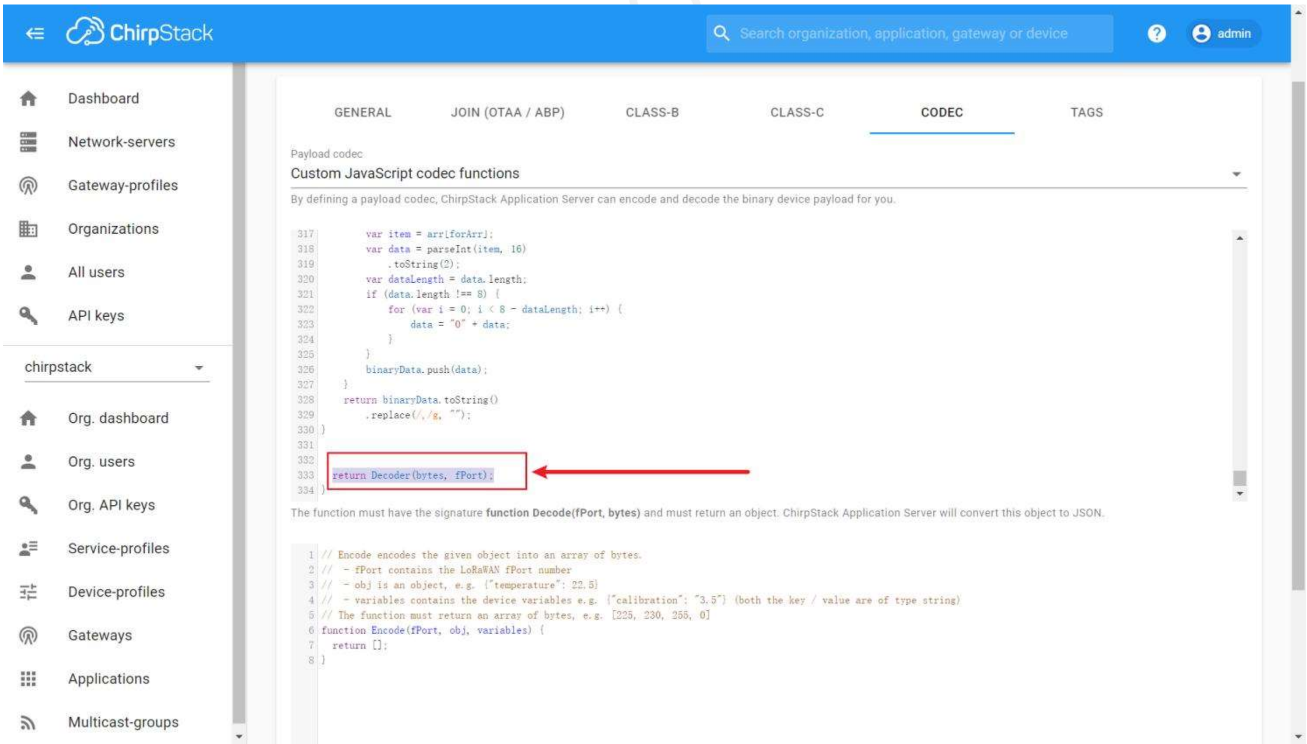
```
// Decode decodes an array of bytes into an object.
//  - fPort contains the LoRaWAN fPort number
//  - bytes is an array of bytes, e.g. [225, 230, 255, 0]
//  - variables contains the device variables e.g. {"calibration": "3.5"} (both the key / value are of type string)
// The function must return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

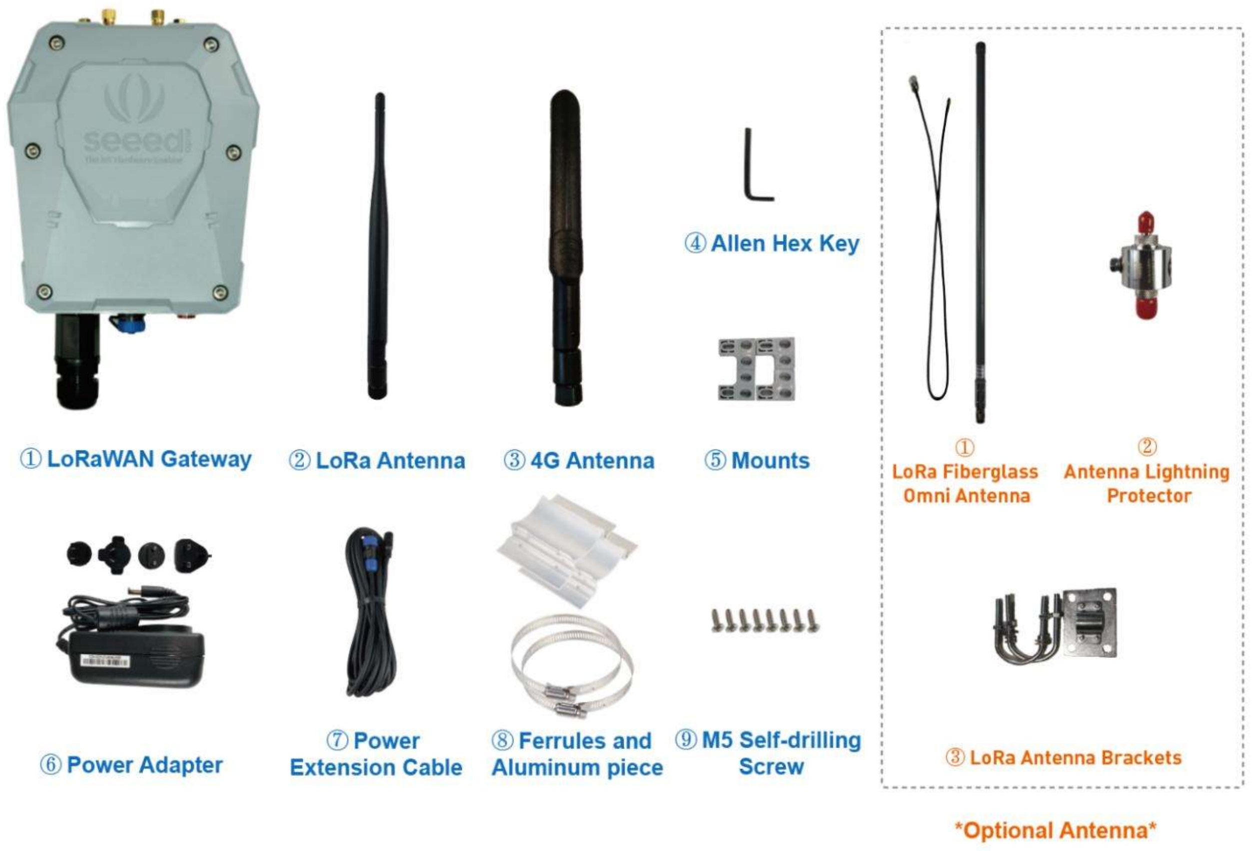
Finally, click “Create”.

5 Device Installation

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

5.1 **Part List**

5.1.1 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.

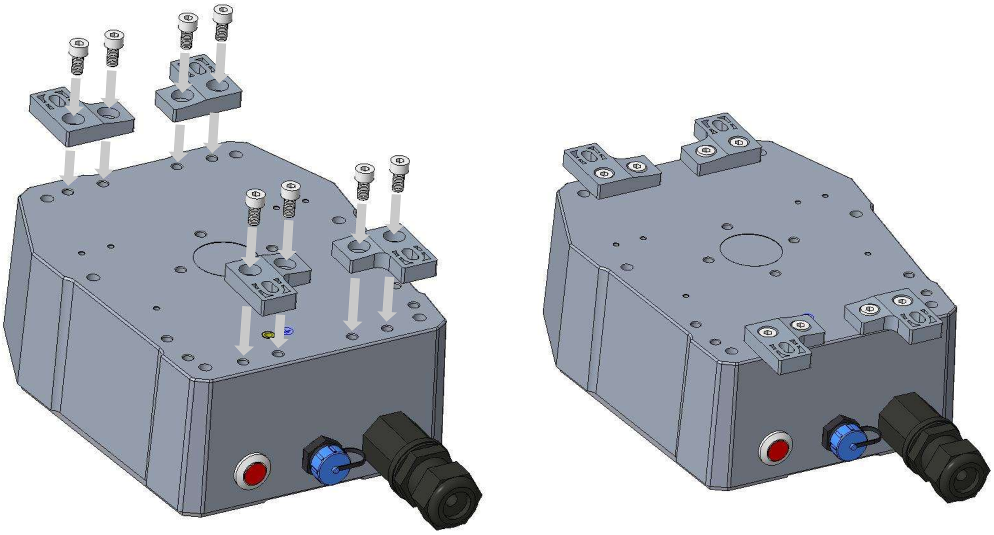
Item	Name	Quantity
1	LoRa Gateway	1
2	LoRa Antenna	1
3	4G Antenna	1
4	Allen Hex Key	1
5	Mounts	4
6	Power Adapter	1
7	Power Extension Cable (5M)	1
8	Ferrules / Aluminum piece	2 / 2
9	M5 Self-drilling Screw	8
10	Antenna Lightning Protector (*Optional)	1
11	LoRa Fiberglass Omni Antenna (*Optional)	1
12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

5.2 Gateway Installation

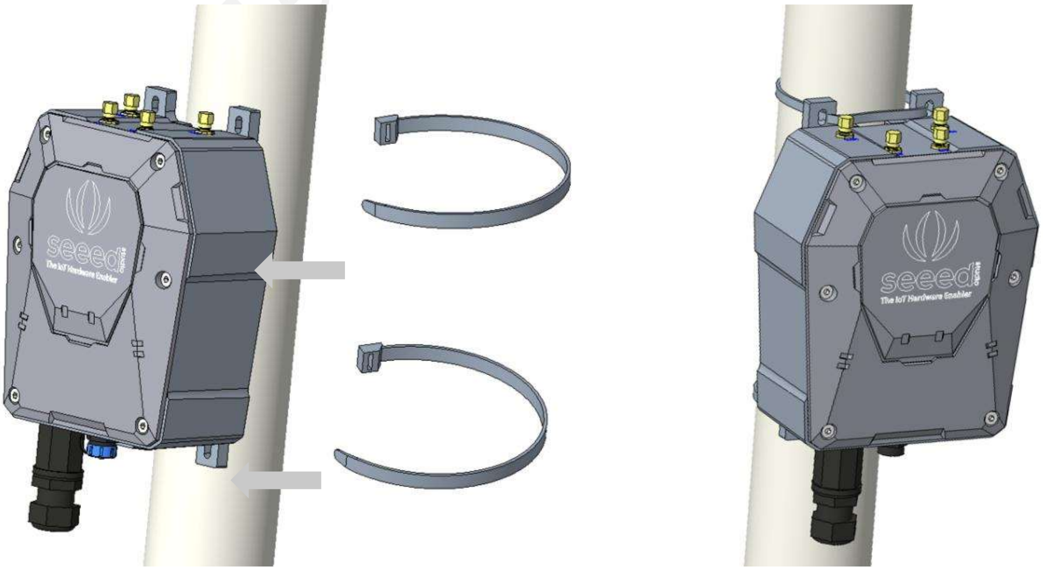
5.2.1 Gateway Installation Methods

- **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.

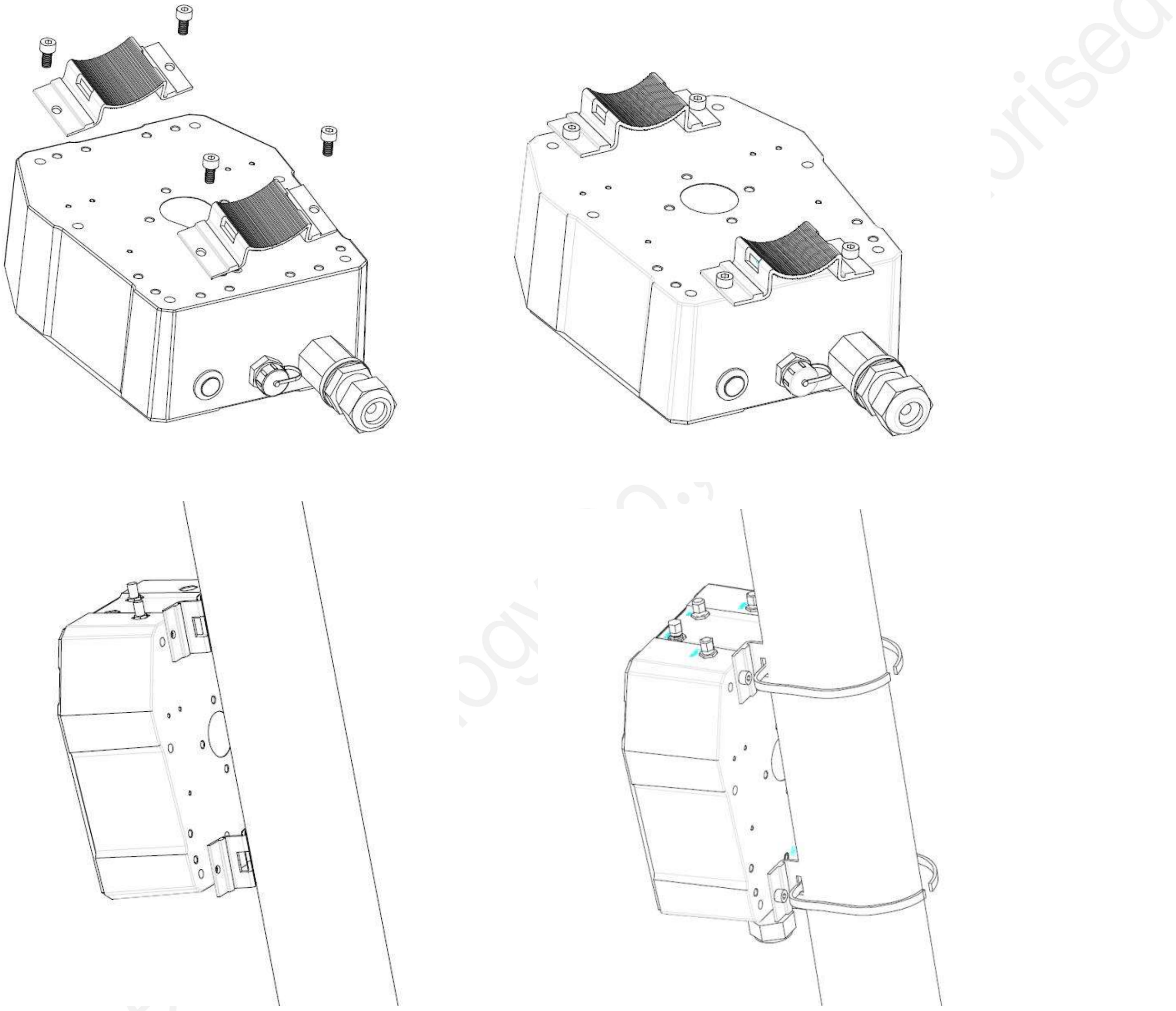


Put cable ties through the holes of the bracket and pull to fasten onto the pole. To get a better communication range, it is recommended to mount the gateway 3 meters above the ground. If there are tall buildings around, the gateway should be kept away from the building or mounted on top of the tall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



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

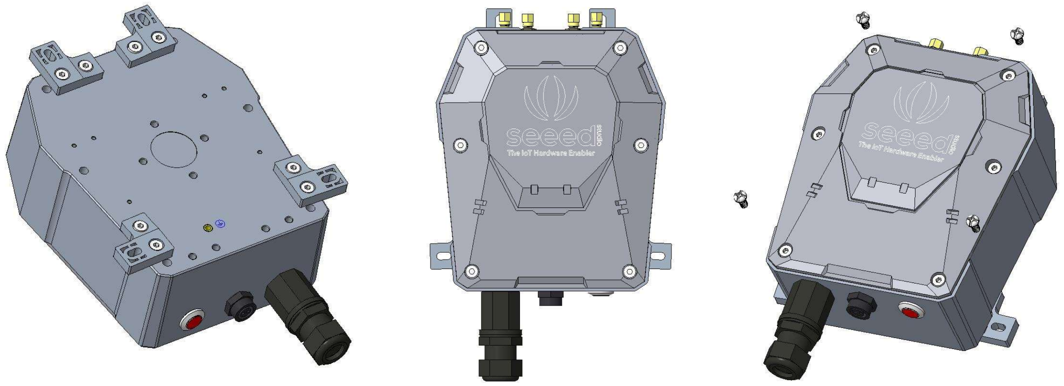
Firstly, use M5 self-drilling screws (included in the package) to fasten the 2 Aluminum pieces onto the gateway. And then use ferrules to fasten the gateway onto the pole. The recommended pole diameter is 76mm.



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.

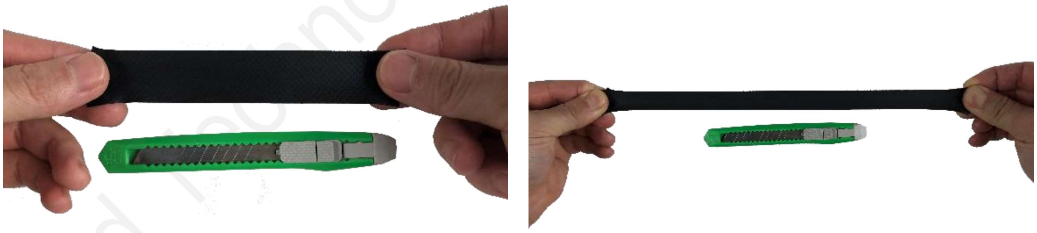


Note: The screws (that fasten gateway onto the wall) are not included in the package. Please prepare screws according to the wall materials (recommended screw diameter: 6mm).

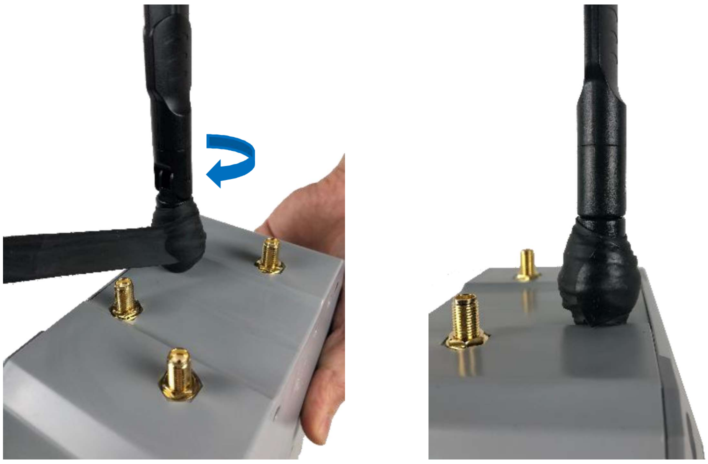
5.2.2 Installation Precautions

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.



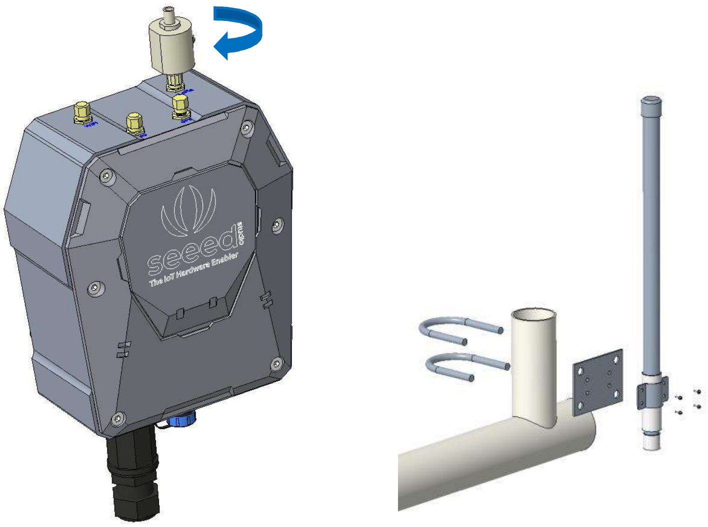
Note: The tape must be wound clockwise because the antenna is fastened clockwise. Otherwise, the antenna may loosen.

If the sensor has wires, install threaded tubes:

5.2.3 Installing Fiberglass LoRa Antenna

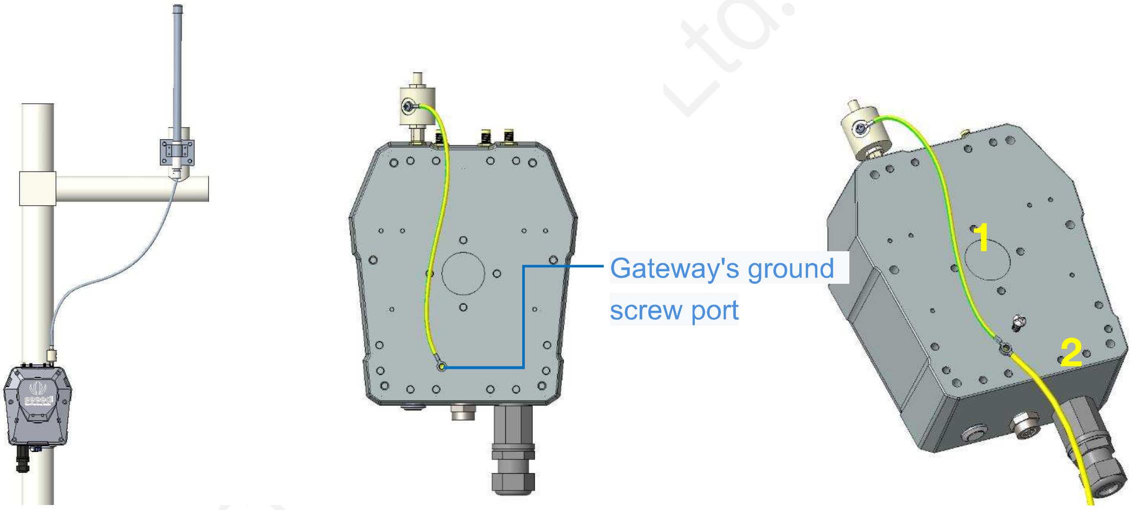
There are two kinds of LoRa antennas: the normal LoRa antenna (included in the package), and the fiberglass LoRa antenna (to be purchased separately). We will introduce how to install the fiberglass LoRa antenna.

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.



5.2.4 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.