

# Get Start with RAK LoRa Develop Kit

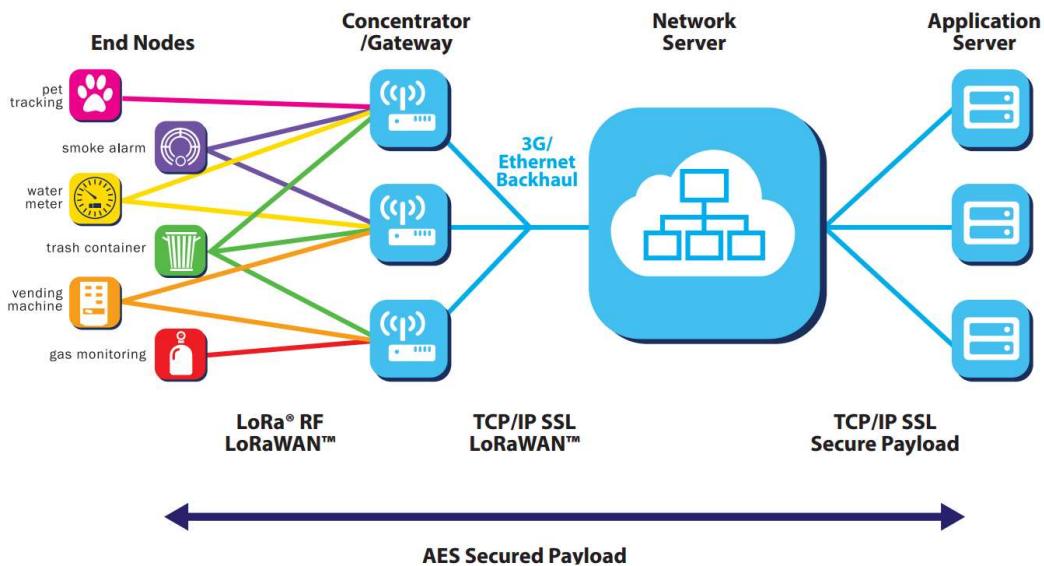
-----The simplest LoRa Kit

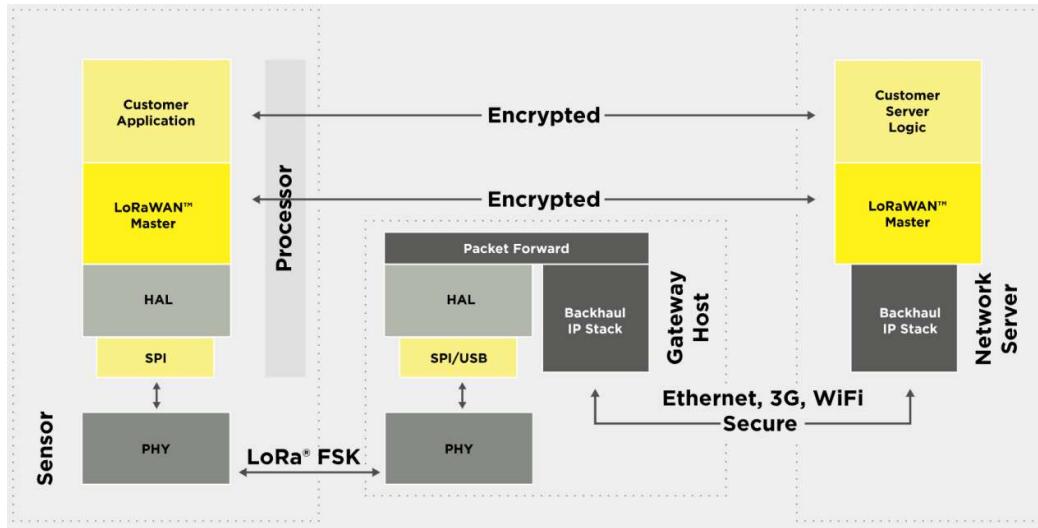
## What's LoRa?

LoRaWAN is a long range, low power wireless protocol that is intended for use in building IoT networks. IoT devices (“nodes”) send small data packets to any number of “gateways” that may be in the several-kilometer range of a node via the LoRaWAN wireless protocol. The gateways then use more traditional communications such as wired Internet connections to forward the messages to a network-server which validates the packets and forwards the application payload to an application-server.

The nature of the LoRa network potentially allows IoT devices to run for years on small batteries, occasionally sending out small packets of data, waiting for a short time for response messages, and then closing the connection until more data needs to be sent. Devices can also be set up so that they are always listening for messages from their applications, though this obviously requires more power and may be more appropriate for devices that are, say, plugged in to a wall socket.

Of course, there is much more to LoRaWAN than is described here. The LoRaWAN protocol is defined and managed by the [LoRa Alliance](#). There is a great deal of information available there.





## About RAK

Shenzhen Rakwireless Technology Co., Ltd. (RAK) was established in June 2014. Based in Shenzhen with a R&D center in Shanghai, a office in Beijing. RAK devoted to developing and supplying advanced IoT technology and services. RAK has become from R&D, production to sales one-stop IoT solution supply company. More information, you can see their official website:

<https://www.rakwireless.com/en/>

## What's the RAK LoRa Develop Kit?

The RAK LoRa Develop Kit is a set of RAK LoRa devices which has been customized for quick start to use. It can be used easily by anyone who want to study, experience, research, develop, test, or deploy LoRa, even if he/she knows nothing about LoRa.

The RAK LoRa Develop Kit includes the following components mainly:

**A RAK LoRa gateway.** It may be RAK2243 + Raspberry Pi;

**A RAK LoRa server OS image.** You can download it from [www.rakwireless.com](http://www.rakwireless.com) by using the verification code which has been placed in the box of RAK LoRa Develop Kit you bought.

**A RAK LoRa node.** It may be RAK811.

**OK, let's go**



## Get start with RAK LoRa Develop Kit

### What do you need to prepare?

1. A set of RAK LoRa Develop Kit.
2. A PC, if you want to use an independent LoRa Server by installing the RAK LoRa server OS image.

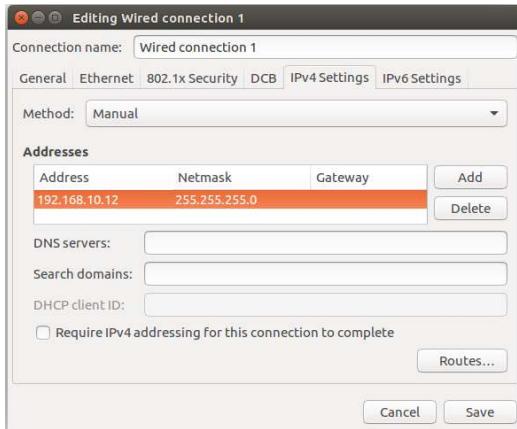
This PC will be used as LoRa Server and Application Server after installing a new system automatically. Please notice that it will erase the old data of this PC.

**Note:** Maybe you can install the system into a virtual machine, but it may lead to an unfriendly user experience, so we don't recommend this way. Just do it as you like.

3. A PC with tools for SSH and AT command.

This PC will be normally used to configure LoRa gateway through SSH, and LoRa node through AT command.

**Note:** The IP address of the LoRa gateway ethernet is 192.168.10.10 by default. You must configure the IP address of your PC to the same IP segment with LoRa gateway. If you don't know how to do it, feel free and just google or Baidu. For example:



## How do you want to use the RAK LoRa Develop Kit?

In this tutorial, there are three cases to use the RAK LoRa Develop Kit:

[Case 1: I know nothing or few things about LoRa, but I want to set up a LoRa network quickly for learning, experiencing, testing, or using, in 3 minutes.](#)

[Case 2: I want to set up a LoRa network quickly with an independent LoRa Server, not an integrated one.](#)

[Case 3: I want to set up a LoRa network quickly, but I want to use TTN as my LoRa server.](#)

OK, Let's have a look at these cases.

**Case 1: I know nothing or few things about LoRa, but I want to set up a LoRa network quickly for learning, experiencing, testing, or using, in 3 minutes.**

Really? in 3 minutes?

Oh, yes! What you read is right! You can set up a LoRa network by yourself in 3 minutes using the RAK LoRa Develop Kit, even if you know nothing about LoRa.

Now, do as the following steps to set up your LoRa network:

**Step 1: Power on the RAK LoRa gateway;**

**Step 2: Send several AT command to the LoRa node from PC as follow:**

- 1) Set the dev\_eui and app\_key, which you'll find in the box of the RAK LoRa Develop Kit or in this tutorial, into the LoRa node through AT command:

```
"at+set_config=dev_eui:1111111111111111"
```

```
"at+set_config=app_eui:70B3D57ED00157D1"
```

```
"at+set_config=app_key:ecbb8ac2bdf563dbe505d2f850219c17"
```

- 2) Join the LoRaWAN network through AT command "at+join=otaa". If you see "at+recv=3,0,0", it indicates that the LoRa node has joined the LoRaWAN network successfully.
- 3) Try to send some data to the application server.

For example:

```
"at+send=0,2,0000000000000007F0000000000000000"
```

**Step 3: Nothing else. That's all! Congratulations! You've set up a LoRa network successfully.**

If you want to check the data on the website of LoRa server and LoRa application server, please do as follow:

- 1) Open the browser and type <http://192.168.10.10:8080>, you will see the login page of LoRa server as follow:

The image shows a login form titled "Login". It contains two input fields: "Username\*" and "Password\*", both preceded by an asterisk indicating they are required. Below the password field is a "LOGIN" button. The entire form is contained within a light gray rectangular box.

Login the LoRa server. The default username and password are both "admin".

- 2) You can see the following page:

The screenshot shows the LoRaServer web interface. On the left, a sidebar lists various management categories: Network-servers, Gateway-profiles, Organizations, All users, loraserver (selected), Org. settings, Org. users, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups. The main content area is titled "Applications". It displays a table with one row, "Box-app", which has a service profile of "box-service-profile" and a description of "data transfer test". A search bar at the top right says "Search organization, application, gateway or device". A "CREATE" button is located in the top right corner of the main content area.

- 3) Enter “Box-app”, there are three pre-configured devices, and you have got their dev\_eui and app\_key from the box of the RAK LoRa Develop Kit or this tutorial:

This screenshot shows the "Applications / Box-app" configuration page. The sidebar is identical to the previous one. The main content is divided into three tabs: "DEVICES" (selected), "APPLICATION CONFIGURATION", and "INTEGRATIONS". The "DEVICES" tab displays a table with three rows: "Rak-node-1" (Device name), "1111111111111111" (Device EUI), "n/a" (Link margin), and "n/a" (Battery). The other two rows, "Rak-node-2" and "Rak-node-3", show similar information. A "DELETE" button is in the top right, and a "CREATE" button is in the bottom right. A search bar at the top right says "Search organization, application, gateway or device".

You can check the status and data of LoRa node by click the “Device name” of LoRa node which you are using on this page.

This screenshot shows the "Applications / Box-app / Devices / Rak-node-1" page. The sidebar is consistent. The main content includes tabs for "CONFIGURATION", "KEYS (OTAA)", "ACTIVATION", "LIVE DEVICE DATA" (selected), and "LIVE LORAWAN FRAMES". Under "LIVE DEVICE DATA", there is a table with five rows of uplink data, each with a dropdown arrow icon. Below the table are buttons for "HELP", "PAUSE", "DOWNLOAD", and "CLEAR". A search bar at the top right says "Search organization, application, gateway or device".

The screenshot shows the LoRaServer web interface. The left sidebar has sections for Network-servers, Gateway-profiles, Organizations, All users, and a dropdown for 'loraserver' which includes Org. settings, Org. users, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups. The main content area shows the path Applications / Box-app / Devices / Rak-node-1. It has tabs for Configuration, Keys (OTAA), Activation, Live Device Data, and Live Lorawan Frames. Below these are buttons for Help, Pause, Download, and Clear. The main panel displays two uplink messages. The first message at 11:52:26 PM is a JSON object with fields like adr, applicationID, applicationName, data, devEUI, deviceName, ICm, IOr, netID, network, gatewayID, infoAttn, location, altitude, latitude, longitude, reason, resI, resInfo, dr, frequency, and dr. The second message at 11:52:23 PM is similar but shorter.

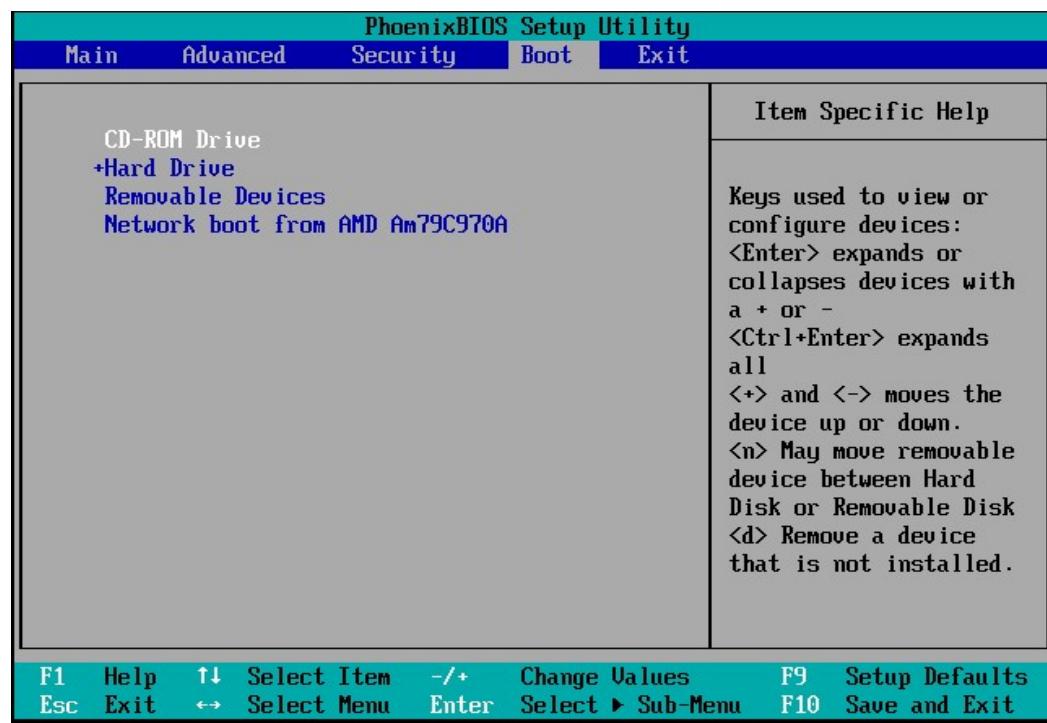
**Note:** There is a RAK LoRa gateway in the RAK LoRa Develop Kit you bought. By default, a LoRa server and a LoRa application server have been integrated into this LoRa gateway too. When you power on this RAK LoRa gateway, it will do some configurations automatically to finish all the steps of LoRa gateway, LoRa server, and LoRa application server. Normally, people call this feature “plus and play”.

## Case 2: I want to set up a LoRa network quickly with an independent LoRa Server, not an integrated one.

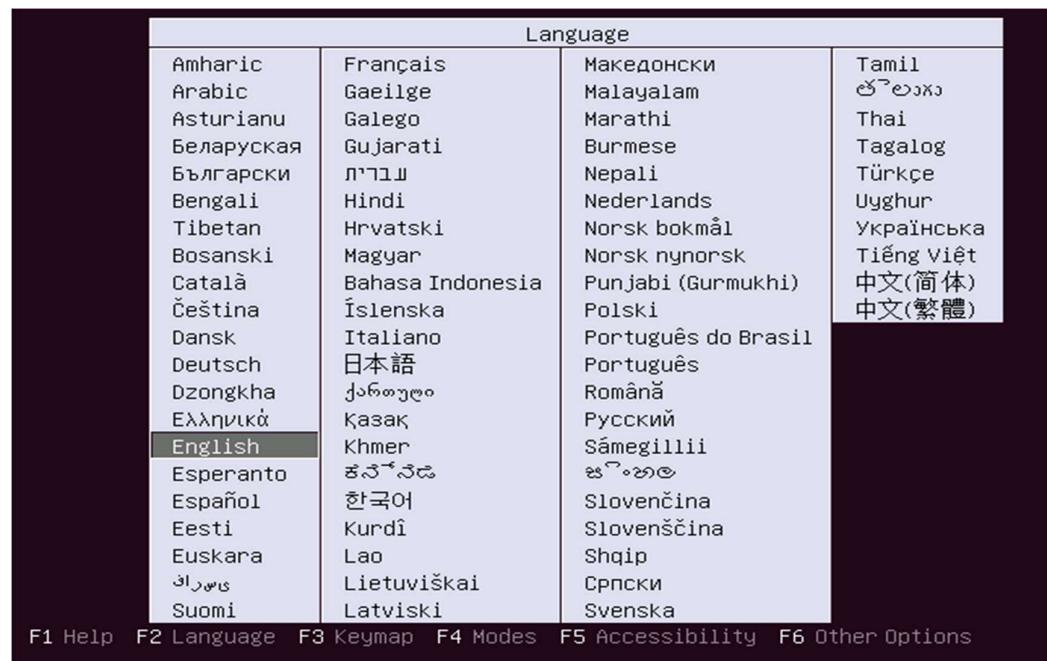
Just do as the following steps:

### Step 1: Build a LoRa server by installing the RAK LoRa Server OS.

- 1) Make a startup disk using the RAK LoRa server OS image into an USB disk or a CD-ROM, if you can't make it, please google or Baidu;
- 2) Install the OS from USB disk or CD-ROM as follow. Actually, it is a normal process of installing an ubuntu OS, so if you have any issues in the process of installing, you can find the solution on google or Baidu;
- 3) Enter BIOS and configure the boot way, “CD-ROM Drive” or “Removable Devices”;



- 4) When beginning to install, it will remind you to choose the language as the following picture, and we recommend you to choose “English”.



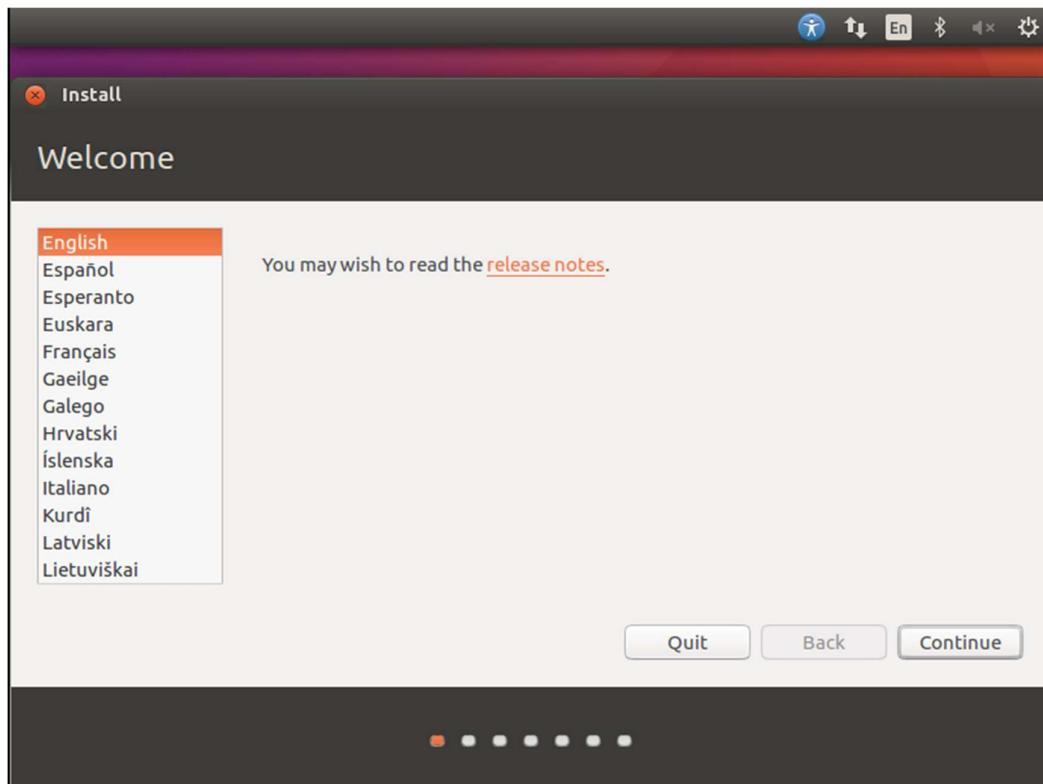
- 5) Select “Install Ubuntu” to install the customized OS which is based on Ubuntu and includes a LoRa server and an Application server. We call it “the LoRa Server OS” in this tutorial.



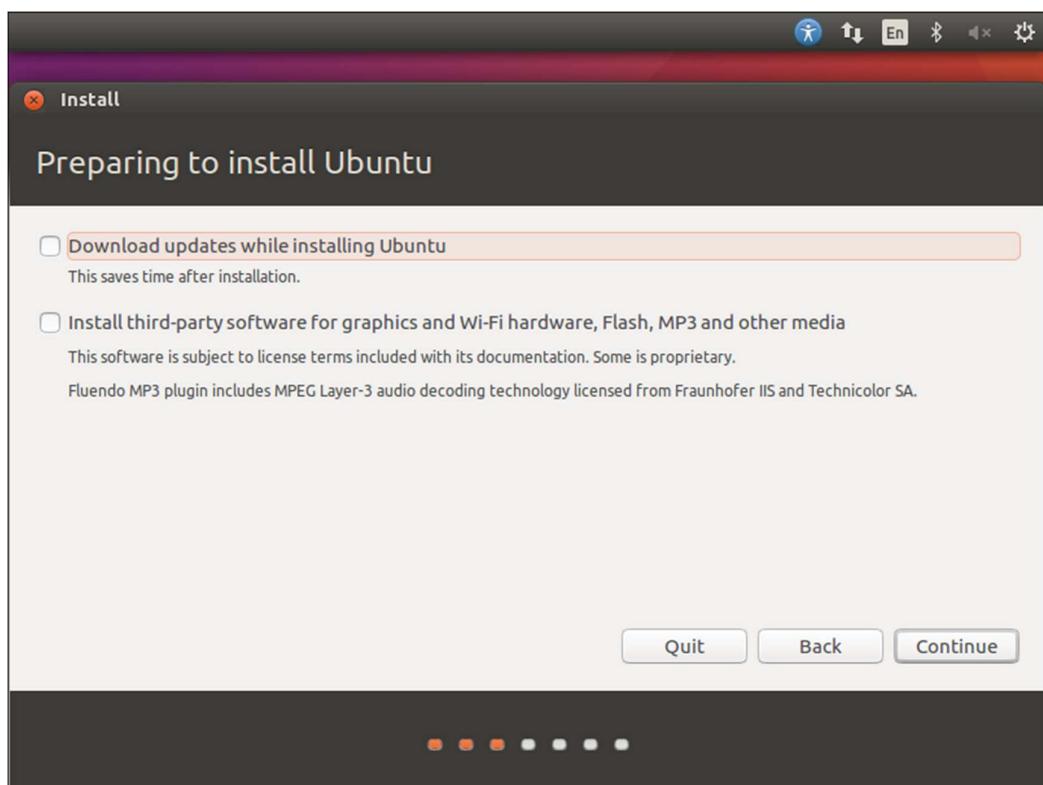
6) Wait for loading...



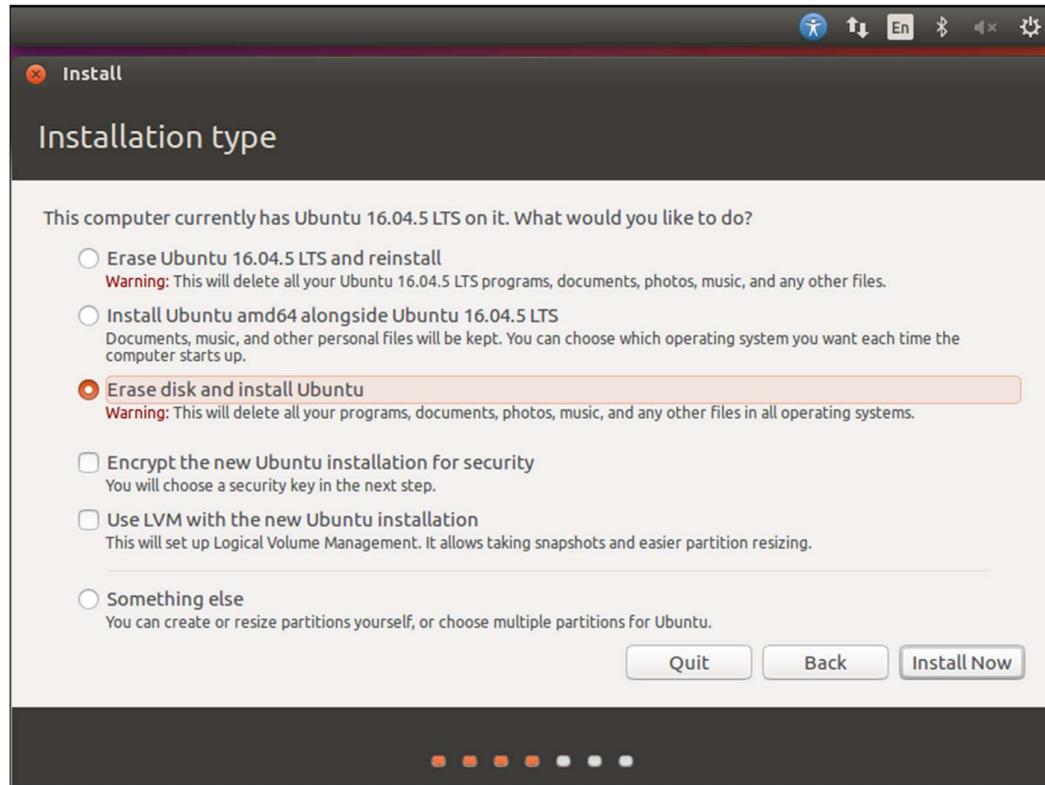
- 7) After loading, select the language again and “Continue” to the next page.



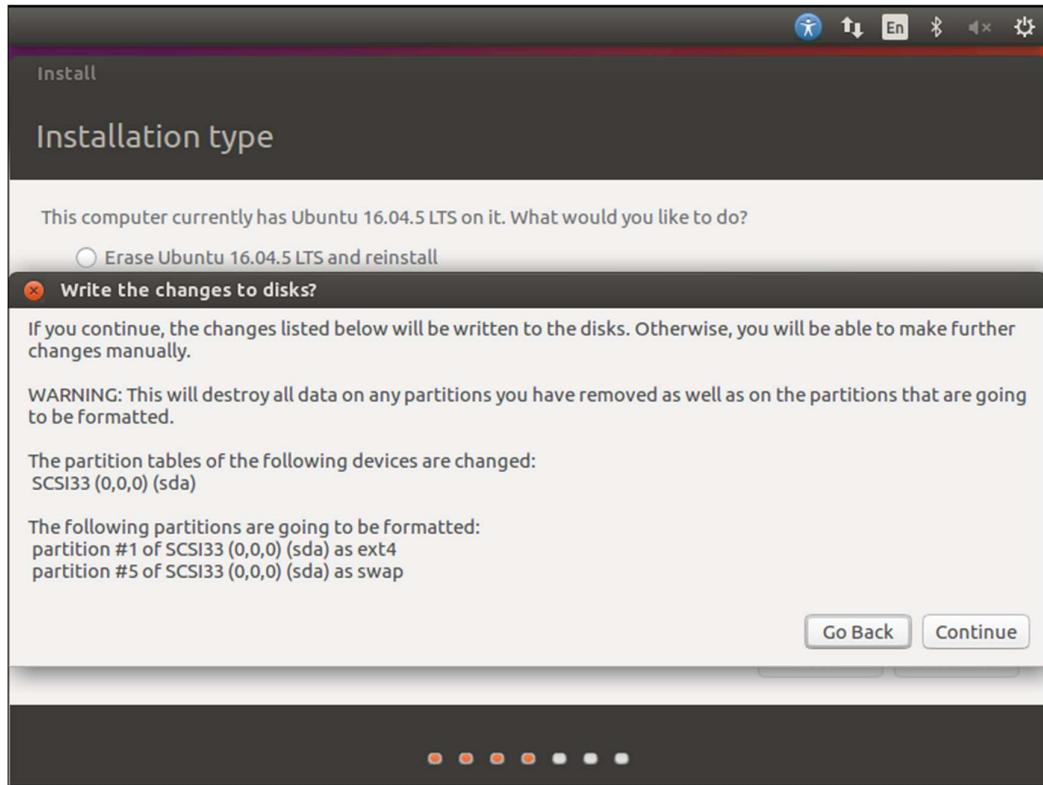
- 8) In this page, please don't select anyone of the check boxes.



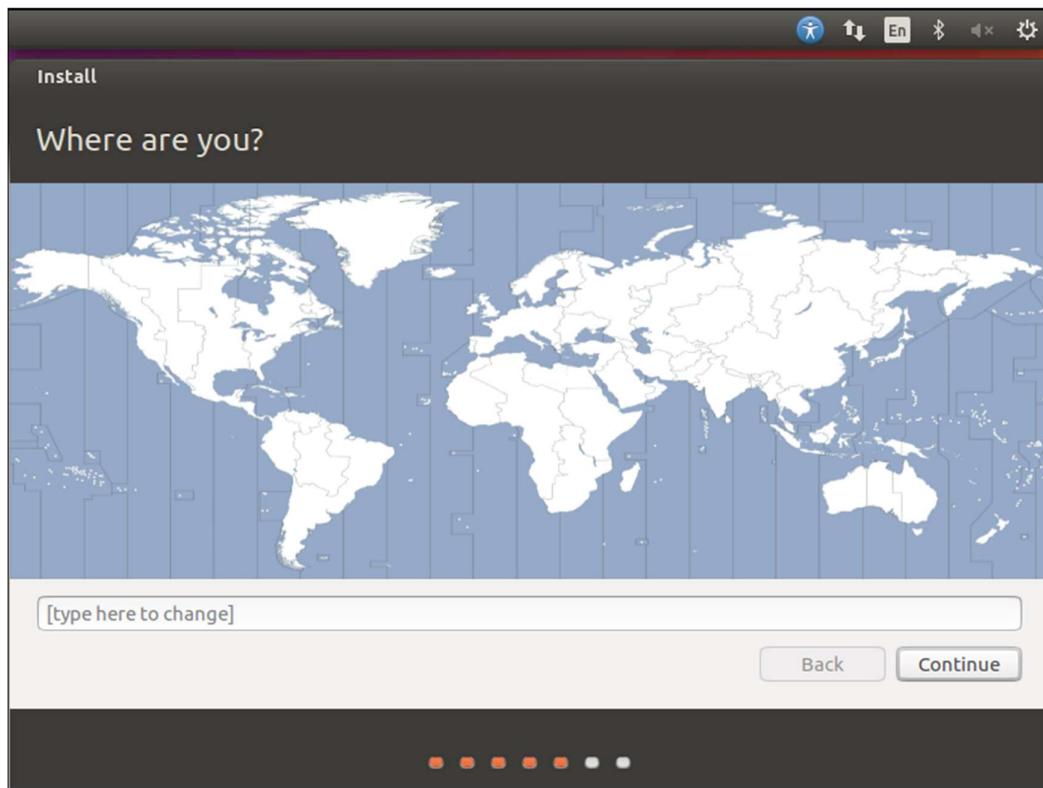
- 9) In the following page, you need to select “Erase disk and install Ubuntu” and click the “Install Now” button, but before that, you should notice that this will delete all your programs, documents, photos, music, and any other files in the old OS if it existed yet.



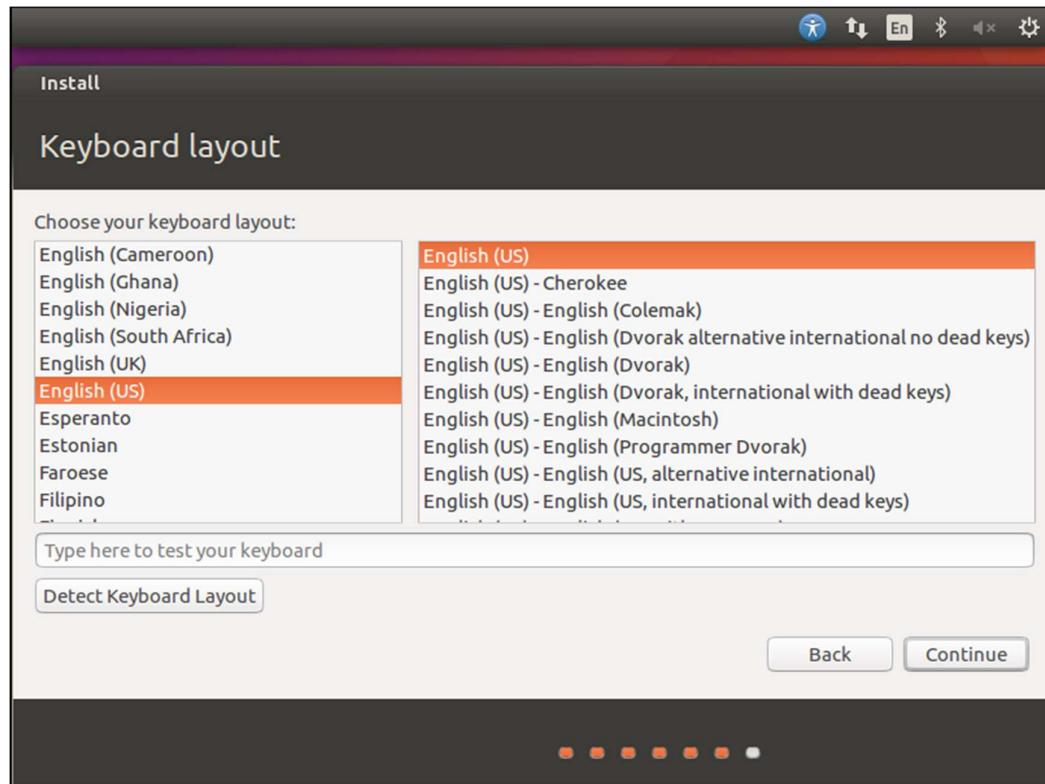
- 10) A pop-up dialog, just click the “Continue” button.



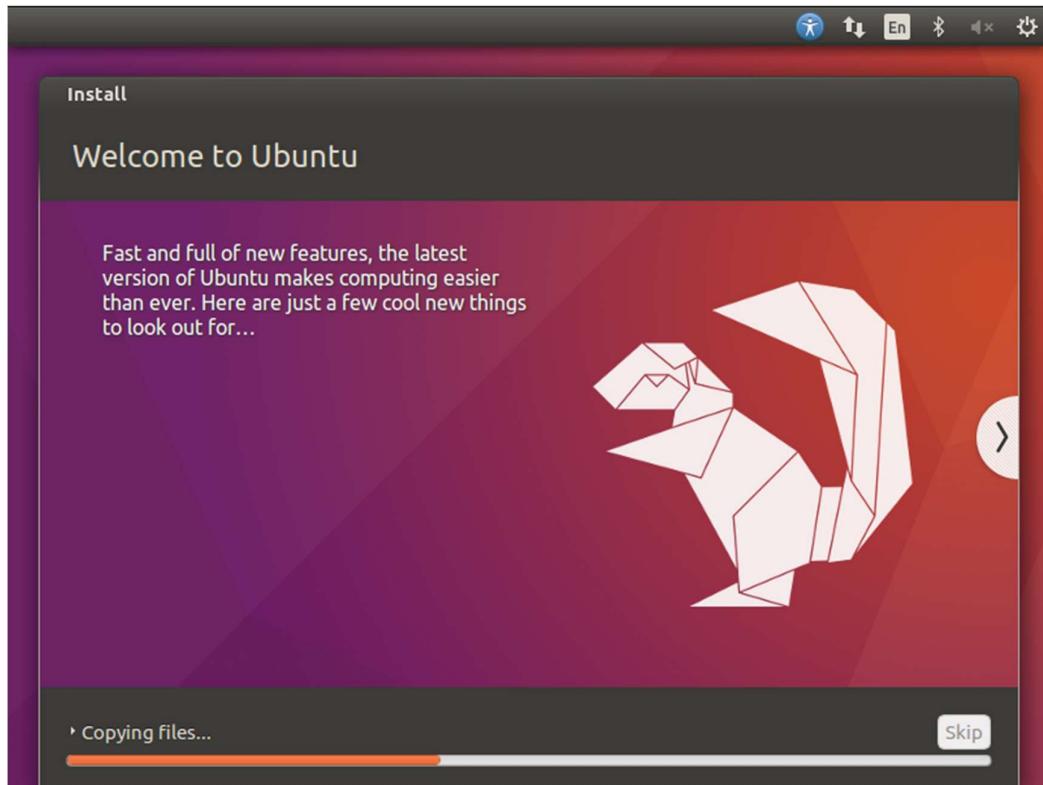
- 11) Select the time zone by typing the name of the time zone or selecting in the map directly. Then “Continue”.



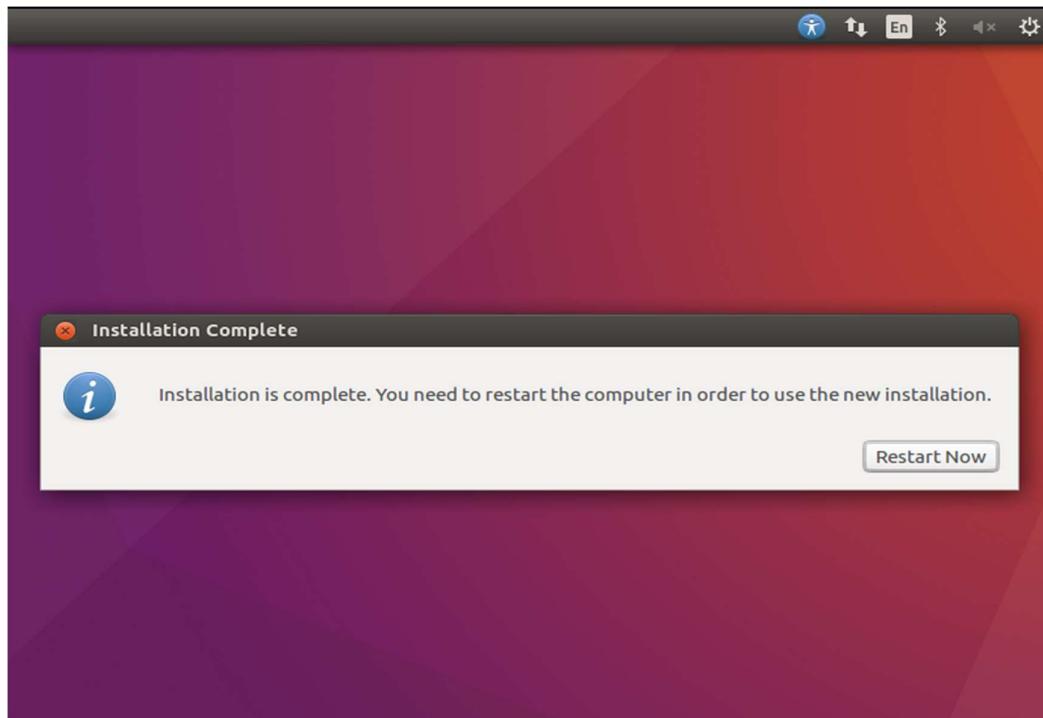
12) Select the keyboard layout as follow.



13) After these steps above, you will see the following page. Now it begin to install the LoRa server OS, and this step will take several minutes or more, just wait patiently.

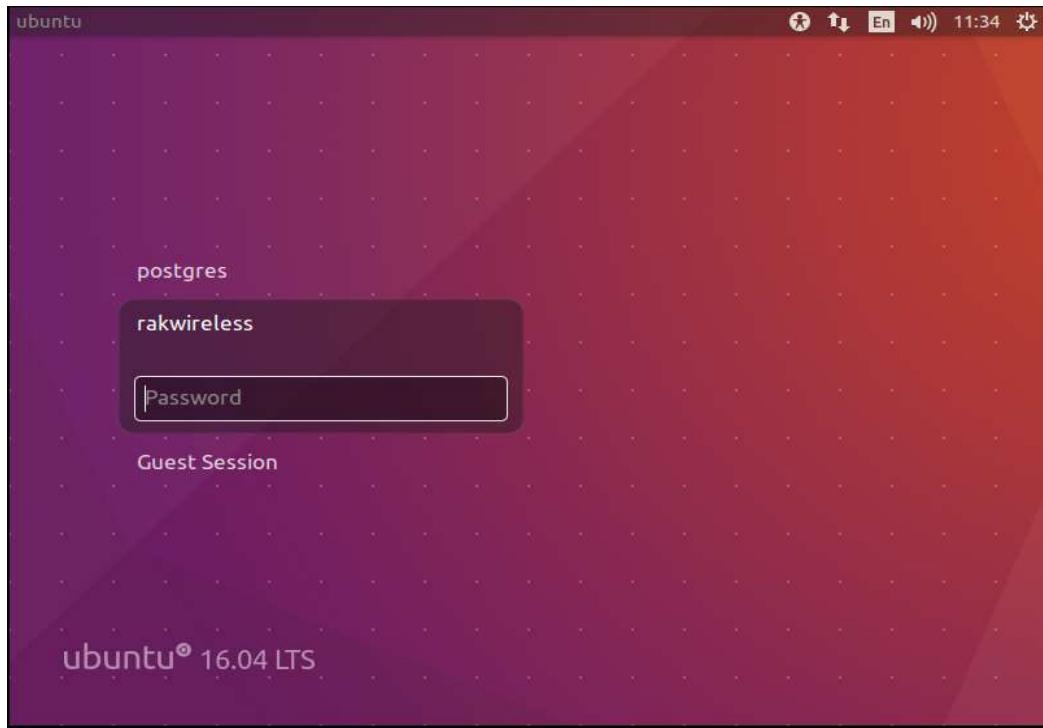


- 14) When it finishes, you should “Restart Now” in the following page.



- 15) After restarting, you will see the login page as follow, it means that you have installed the LoRa Server OS successfully! Just select the user “rakwireless”

and type the default password “rakwireless” to enter the OS.



- 16) Now, Configure the IP address of your LoRa server, and remember it because you will use it in the next step when configuring your LoRa gateway.
- 17) What? Don't know how to configure the IP address? Just google or Baidu “How to configure IP address for Ubuntu”.

## Step 2: Configure your LoRa gateway

There are some kinds of RAK LoRa gateway, e.g. RAK831, RAK833, RAK2243, RAK2247, etc. You may have anyone of them.

In this tutorial, we'll use “RAK2243 + Raspberry Pi + LoRa gateway OS” which have been placed into the box of the RAK LoRa develop Kit you bought as our LoRa gateway to show how to configure it.

Just do as follow:

- 1) Login the Raspberry Pi through SSH, the default username and password are both “admin”.

Note: By default, the IP address of the LoRa gateway ethernet is 192.168.10.10, and you can connect to it through wire using a PC and SSH.

For example:

```
jimmy@jimmy-Rak: ~
jimmy@jimmy-Rak:~$ ssh admin@192.168.10.10
```

```
jimmy@jimmy-Rak: ~
jimmy@jimmy-Rak:~$ ssh admin@192.168.10.10
The authenticity of host '192.168.10.10 (192.168.10.10)' can't be established.
RSA key fingerprint is SHA256:qf+U1erxKMUOckmyKkiCPLTFq4rPX2NwrUb4WM7XwKQ.
Are you sure you want to continue connecting (yes/no)?
```

```
jimmy@jimmy-Rak: ~
jimmy@jimmy-Rak:~$ ssh admin@192.168.10.10
The authenticity of host '192.168.10.10 (192.168.10.10)' can't be established.
RSA key fingerprint is SHA256:qf+U1erxKMUOckmyKkiCPLTFq4rPX2NwrUb4WM7XwKQ.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/jimmy/.ssh/known_hosts)
.
admin@192.168.10.10's password:
```

**Note:** If you don't know how to use SSH, please google or Baidu.

- 2) After that, you will see the following page, which indicates you have login successfully.

```
jimmy@jimmy-Rak:~  
jimmy@jimmy-Rak:~$ ssh admin@192.168.10.10  
The authenticity of host '192.168.10.10 (192.168.10.10)' can't be established.  
RSA key fingerprint is SHA256:qf+U1erxKMU0ckmyKkiCPLTFq4rPX2NwrUb4WM7XwKQ.  
Are you sure you want to continue connecting (yes/no)? yes  
Failed to add the host to the list of known hosts (/home/jimmy/.ssh/known_hosts)  
.admin@192.168.10.10's password:  
_____  
documentation and copyright information: www.loraserver.io  
  
Commands:  
> sudo gateway-config - configure the gateway  
> sudo monit status - display service monitor  
raspberrypi3:~$
```

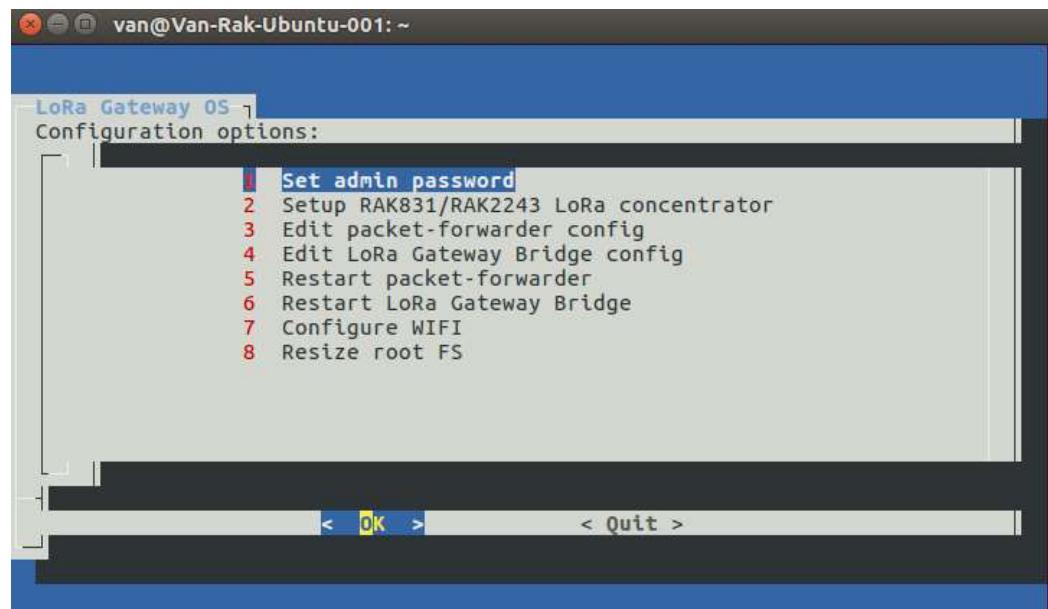
- 3) Enter the command “sudo gateway-config” and the login password to configure the gateway. This command has been shown in the page.

```
jimmy@jimmy-Rak:~  
jimmy@jimmy-Rak:~$ ssh admin@192.168.10.10  
The authenticity of host '192.168.10.10 (192.168.10.10)' can't be established.  
RSA key fingerprint is SHA256:qf+U1erxKMU0ckmyKkiCPLTFq4rPX2NwrUb4WM7XwKQ.  
Are you sure you want to continue connecting (yes/no)? yes  
Failed to add the host to the list of known hosts (/home/jimmy/.ssh/known_hosts)  
.admin@192.168.10.10's password:  
_____  
documentation and copyright information: www.loraserver.io  
  
Commands:  
> sudo gateway-config - configure the gateway  
> sudo monit status - display service monitor  
raspberrypi3:~$
```

```
jimmy@jimmy-Rak:~  
jimmy@jimmy-Rak:~$ ssh admin@192.168.10.10  
The authenticity of host '192.168.10.10 (192.168.10.10)' can't be established.  
RSA key fingerprint is SHA256:qf+U1erxKMU0ckmyKkiCPLTFq4rPX2NwrUb4WM7XwKQ.  
Are you sure you want to continue connecting (yes/no)? yes  
Failed to add the host to the list of known hosts (/home/jimmy/.ssh/known_hosts)  
.admin@192.168.10.10's password:  
_____  
documentation and copyright information: www.loraserver.io  
  
Commands:  
> sudo gateway-config - configure the gateway  
> sudo monit status - display service monitor  
raspberrypi3:~$ sudo gateway-config
```

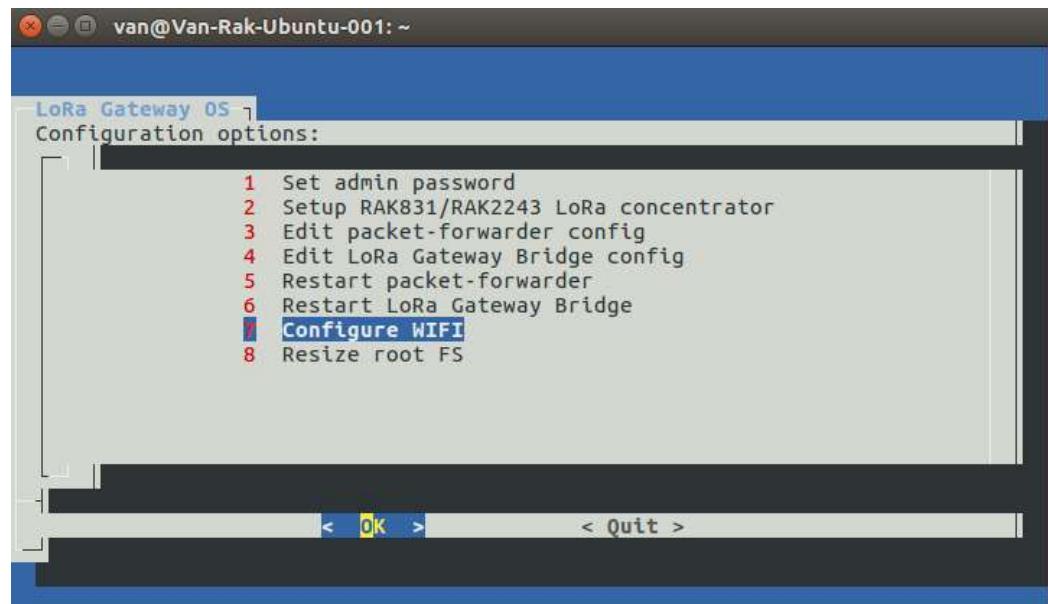
```
jimmy@jimmy-Rak:~  
admin@192.168.10.10's password:  
_____  
documentation and copyright information: www.loraserver.io  
  
Commands:  
> sudo gateway-config - configure the gateway  
> sudo monit status - display service monitor  
raspberrypi3:~$ sudo gateway-config  
  
We trust you have received the usual lecture from the local System  
Administrator. It usually boils down to these three things:  
  
#1) Respect the privacy of others.  
#2) Think before you type.  
#3) With great power comes great responsibility.  
Password: ■
```

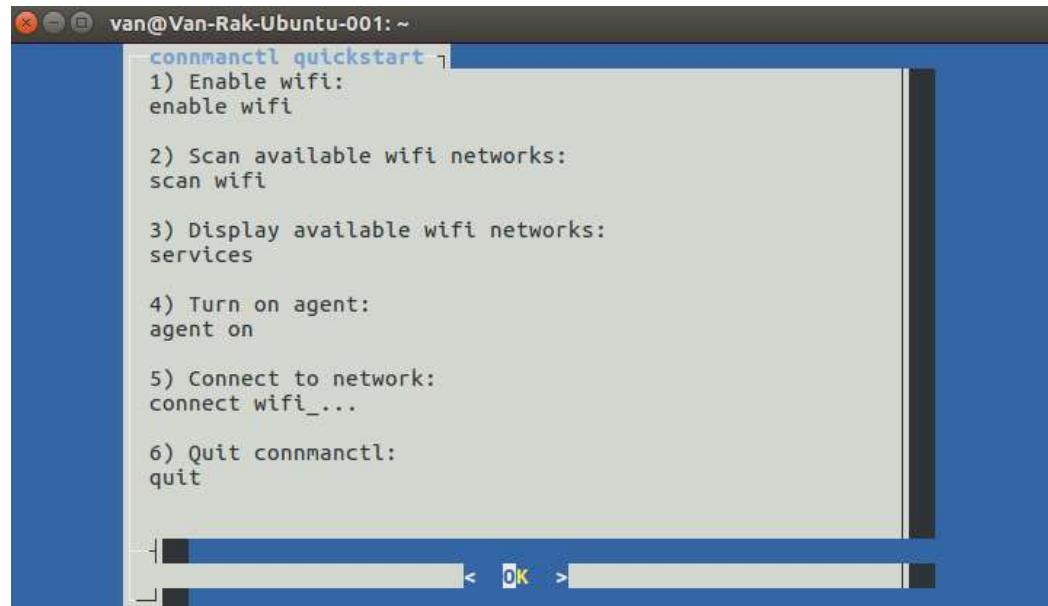
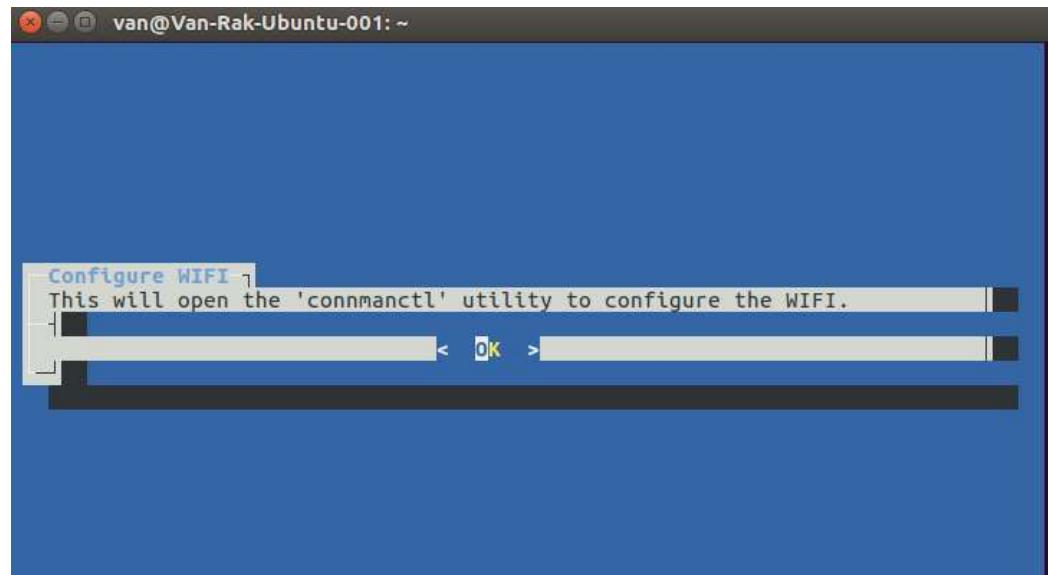
4) Now, you should see a configuration page as follow:



**Note:** In this tutorial, you may only use the option 1, 2, 3, and 7.

- 5) If you want to connect the LoRa gateway to a WiFi network, just do as follow:

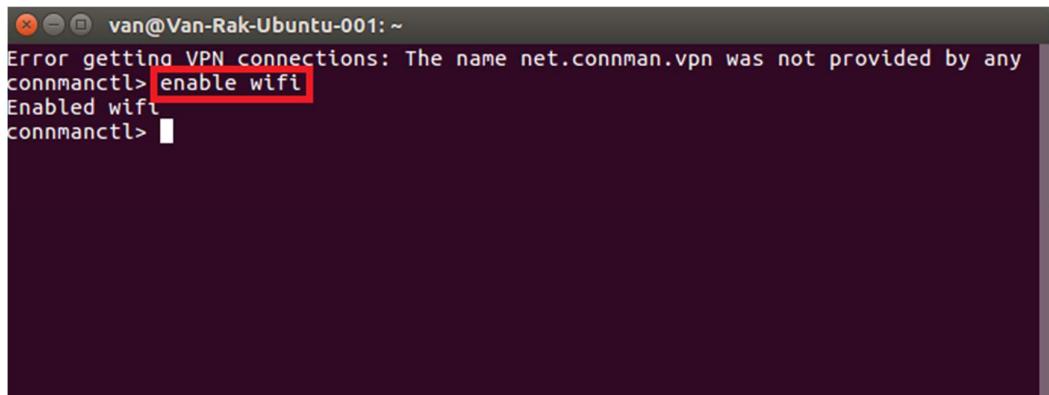






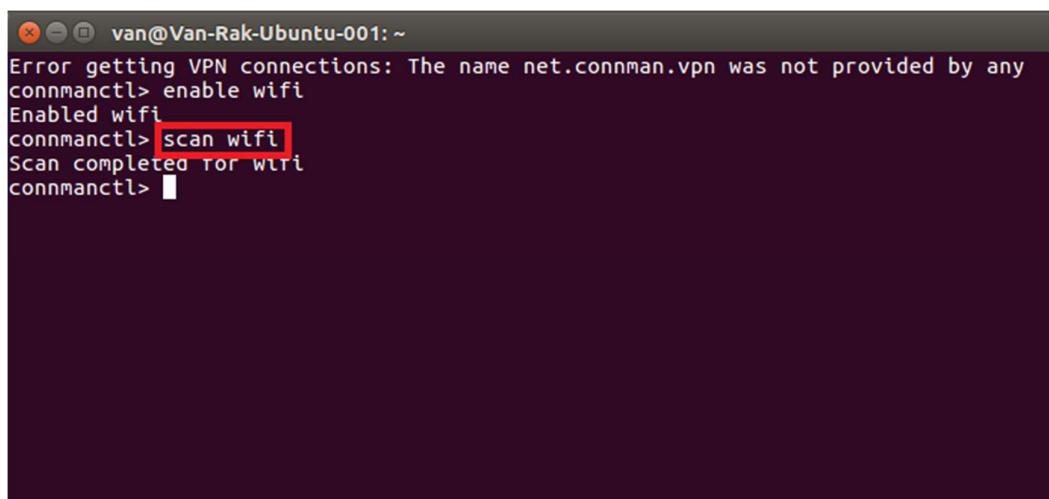
```
van@Van-Rak-Ubuntu-001:~  
Error getting VPN connections: The name net.connman.vpn was not provided by any  
connmanctl> █
```

Enable WiFi:



```
van@Van-Rak-Ubuntu-001:~  
Error getting VPN connections: The name net.connman.vpn was not provided by any  
connmanctl> enable wifi  
Enabled wifi  
connmanctl> █
```

Scan WiFi:



```
van@Van-Rak-Ubuntu-001:~  
Error getting VPN connections: The name net.connman.vpn was not provided by any  
connmanctl> enable wifi  
Enabled wifi  
connmanctl> scan wifi  
Scan completed for wifi  
connmanctl> █
```

Check services:

```
van@Van-Rak-Ubuntu-001:~  
Error getting VPN connections: The name net.connman.vpn was not provided by any  
connmanctl> enable wifi  
Enabled wifi  
connmanctl> scan wifi  
Scan completed for wifi  
connmanctl> services  
*AR Wired          ethernet_b827ebe8a9f8_cable  
                  AccessMe      wifi_b827ebbdxfcad_4163636573734d65_managed_psk  
                  AccessMeTest   wifi_b827ebbdxfcad_4163636573734d6554657374_managed_psk  
                  SENDCOM      wifi_b827ebbdxfcad_53454e44434f4d_managed_psk  
                  WENSEN       wifi_b827ebbdxfcad_57454e53454e_managed_psk  
                  SENFA        wifi_b827ebbdxfcad_53454e4641_managed_psk  
                  sxxxnyjk     wifi_b827ebbdxfcad_737878786e796a6b_managed_psk  
                  HWJD         wifi_b827ebbdxfcad_48574a44_managed_psk  
                  SXRKKG       wifi_b827ebbdxfcad_5358524b474b_managed_psk  
                  XAYNKJ       wifi_b827ebbdxfcad_5841594e4b4a_managed_psk  
                  HT-WIFI      wifi_b827ebbdxfcad_48542d57494649_managed_psk  
connmanctl> ■
```

Agent on:

```
van@Van-Rak-Ubuntu-001:~  
Error getting VPN connections: The name net.connman.vpn was not provided by any  
connmanctl> enable wifi  
Enabled wifi  
connmanctl> scan wifi  
Scan completed for wifi  
connmanctl> services  
*AR Wired          ethernet_b827ebe8a9f8_cable  
                  AccessMe      wifi_b827ebbdxfcad_4163636573734d65_managed_psk  
                  AccessMeTest   wifi_b827ebbdxfcad_4163636573734d6554657374_managed_psk  
                  SENDCOM      wifi_b827ebbdxfcad_53454e44434f4d_managed_psk  
                  SENFA        wifi_b827ebbdxfcad_53454e4641_managed_psk  
                  WENSEN       wifi_b827ebbdxfcad_57454e53454e_managed_psk  
                  SXRKKG       wifi_b827ebbdxfcad_5358524b474b_managed_psk  
                  HWJD         wifi_b827ebbdxfcad_48574a44_managed_psk  
                  TAINENG      wifi_b827ebbdxfcad_5441494e454e47_managed_psk  
                  XAYNKJ       wifi_b827ebbdxfcad_5841594e4b4a_managed_psk  
                  HT-WIFI      wifi_b827ebbdxfcad_48542d57494649_managed_psk  
connmanctl> agent on  
Agent registered  
connmanctl> ■
```

Connect WiFi as follows (notice what you need to enter isn't the SSID):

```

van@Van-Rak-Ubuntu-001: ~
Error getting VPN connections: The name net.connman.vpn was not provided by any
connmanctl> enable wifi
Enabled wifi
connmanctl> scan wifi
Scan completed for wifi
connmanctl> services
*AR Wired          ethernet_b827ebe8a9f8_cable
                  AccessMe          wifi_b827ebbdxfcad_4163636573734d65_managed_psk
                  AccessMeTest        wifi_b827ebbdxfcad_4163636573734d6554657374_managed_psk
                  SENDCOM           wifi_b827ebbdxfcad_53454e44434f4d_managed_psk
                  SENFA              wifi_b827ebbdxfcad_53454e4641_managed_psk
                  WENSEN             wifi_b827ebbdxfcad_57454e53454e_managed_psk
                  SXRKGK            wifi_b827ebbdxfcad_5358524b474b_managed_psk
                  HWJD               wifi_b827ebbdxfcad_48574a44_managed_psk
                  TAINENG            wifi_b827ebbdxfcad_5441494e454e47_managed_psk
                  XAYNKJ             wifi_b827ebbdxfcad_5841594e4b4a_managed_psk
                  HT-WIFI            wifi_b827ebbdxfcad_48542d57494649_managed_psk
connmanctl> agent on
Agent registered
connmanctl> connect wifi_b827ebbdxfcad_4163636573734d65_managed_psk
Agent RequestInput wifi_b827ebbdxfcad_4163636573734d65_managed_psk
  Passphrase = [ Type=psk, Requirement=mandatory ]
Passphrase? ■

```

After entering the password, if you see “connected ...”, it indicates the LoRa gateway has connected WiFi successfully.

```

van@Van-Rak-Ubuntu-001: ~
Enabled wifi
connmanctl> scan wifi
Scan completed for wifi
connmanctl> services
*AR Wired          ethernet_b827ebe8a9f8_cable
                  AccessMe          wifi_b827ebbdxfcad_4163636573734d65_managed_psk
                  AccessMeTest        wifi_b827ebbdxfcad_4163636573734d6554657374_managed_psk
                  SENDCOM           wifi_b827ebbdxfcad_53454e44434f4d_managed_psk
                  SENFA              wifi_b827ebbdxfcad_53454e4641_managed_psk
                  WENSEN             wifi_b827ebbdxfcad_57454e53454e_managed_psk
                  sxxxnyjk          wifi_b827ebbdxfcad_737878786e796a6b_managed_psk
                  SXRKGK            wifi_b827ebbdxfcad_5358524b474b_managed_psk
                  HWJD               wifi_b827ebbdxfcad_48574a44_managed_psk
                  HT-WIFI            wifi_b827ebbdxfcad_48542d57494649_managed_psk
                  XAYNKJ             wifi_b827ebbdxfcad_5841594e4b4a_managed_psk
                  TAINENG            wifi_b827ebbdxfcad_5441494e454e47_managed_psk
connmanctl> agent on
Agent registered
connmanctl> connect wifi_b827ebbdxfcad_4163636573734d65_managed_psk
Agent RequestInput wifi_b827ebbdxfcad_4163636573734d65_managed_psk
  Passphrase = [ Type=psk, Requirement=mandatory ]
Passphrase? rakwireless.2018
Connected wifi_b827ebbdxfcad_4163636573734d65_managed_psk
connmanctl> ■

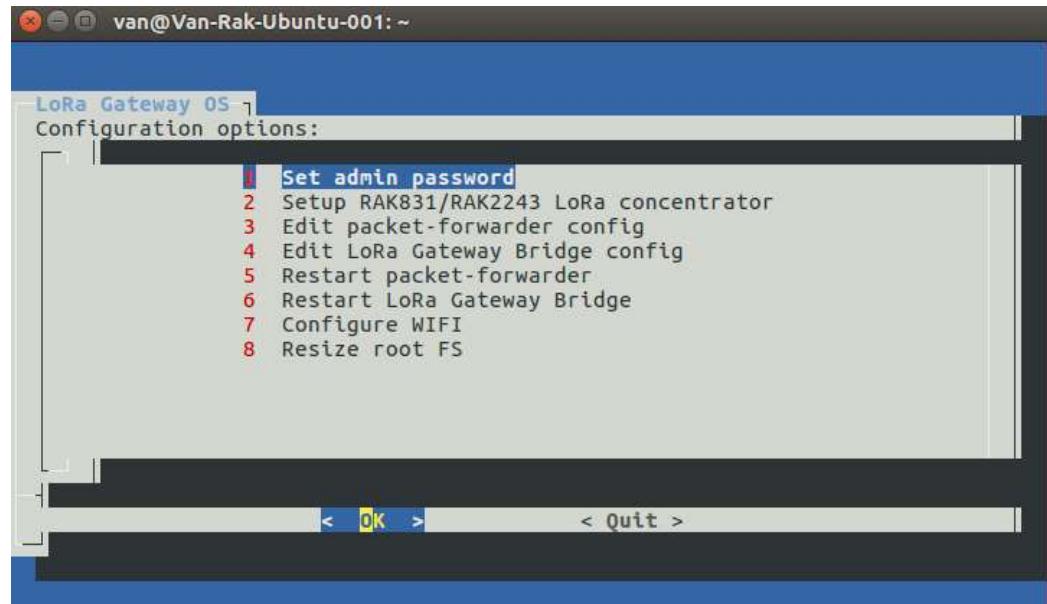
```

Then quit the WiFi configuration page:

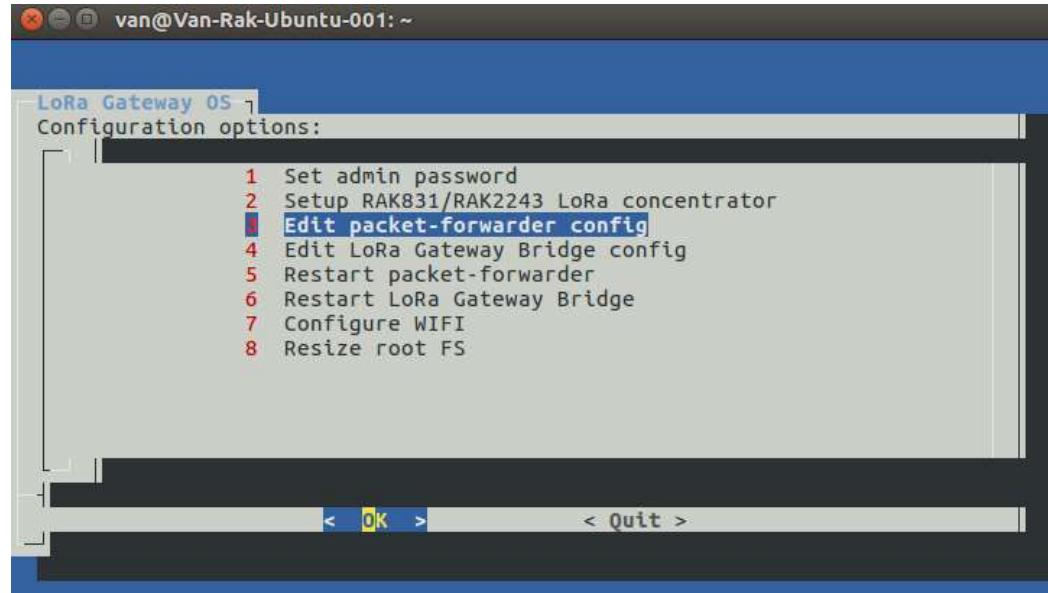
```

van@Van-Rak-Ubuntu-001: ~
Enabled wifi
connmanctl> scan wifi
Scan completed for wifi
connmanctl> services
*AR Wired          ethernet_b827ebe8a9f8_cable
AccessMe           wifi_b827ebbdfcad_4163636573734d65_managed_psk
AccessMeTest       wifi_b827ebbdfcad_4163636573734d6554657374_managed_psk
SENCOM             wifi_b827ebbdfcad_53454e44434f4d_managed_psk
SENFA              wifi_b827ebbdfcad_53454e4641_managed_psk
WENSEN             wifi_b827ebbdfcad_57454e53454e_managed_psk
sxxxnyjk          wifi_b827ebbdfcad_737878786e796a6b_managed_psk
SRKGK              wifi_b827ebbdfcad_5358524b474b_managed_psk
HWJD               wifi_b827ebbdfcad_48574a44_managed_psk
HT-WIFI            wifi_b827ebbdfcad_48542d57494649_managed_psk
XAYNKJ             wifi_b827ebbdfcad_5841594e4b4a_managed_psk
TAINENG            wifi_b827ebbdfcad_5441494e454e47_managed_psk
connmanctl> agent on
Agent registered
connmanctl> connect wifi_b827ebbdfcad_4163636573734d65_managed_psk
Agent RequestInput wifi_b827ebbdfcad_4163636573734d65_managed_psk
  Passphrase = [ Type=psk, Requirement=mandatory ]
Passphrase? rakwireless.2018
Connected wifi_b827ebbdfcad_4163636573734d65_managed_psk
connmanctl> quit

```



- 6) When you power on this LoRa gateway firstly, it works on EU868(EU\_863\_870) and points to the internal LoRa server by default. If you want to check the detail, you can select the option 3



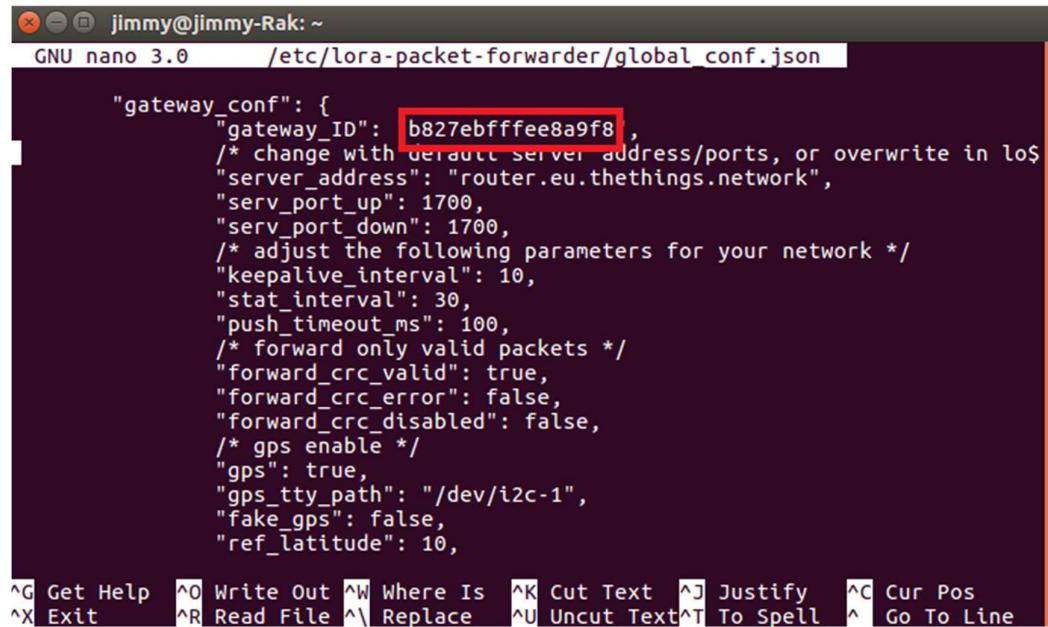
7) Now, you can see the detail of the “global\_conf.json” file:

A screenshot of a terminal window titled "jimmy@jimmy-Rak: ~". The title bar also shows "GNU nano 3.0 /etc/lora-packet-forwarder/global\_conf.json". The file content is displayed in JSON format:

```
{  
    "SX1301_conf": {  
        "lorawan_public": true,  
        "clksrc": 1,  
        "antenna_gain": 0,  
        "radio_0": {  
            "enable": true,  
            "type": "SX1257",  
            "freq": 867500000,  
            "rssи_offset": -166.0,  
            "tx_enable": true,  
            "tx_freq_min": 863000000,  
            "tx_freq_max": 870  
        },  
        "radio_1": {  
            "enable": true,  
            "type": "SX1257",  
            "freq": 868500000,  
            "rssи_offset": -166.0  
        }  
    }  
}
```

The status bar at the bottom indicates "[ Read 171 lines ]". Below the status bar are nano editor key bindings: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^C Cur Pos, ^X Exit, ^R Read File, ^\ Replace, ^U Uncut Text, ^T To Spell, ^ Go To Line.

**Note:** Do you see the gateway ID in the end of this file as shown as the following picture? It is important to remember it!



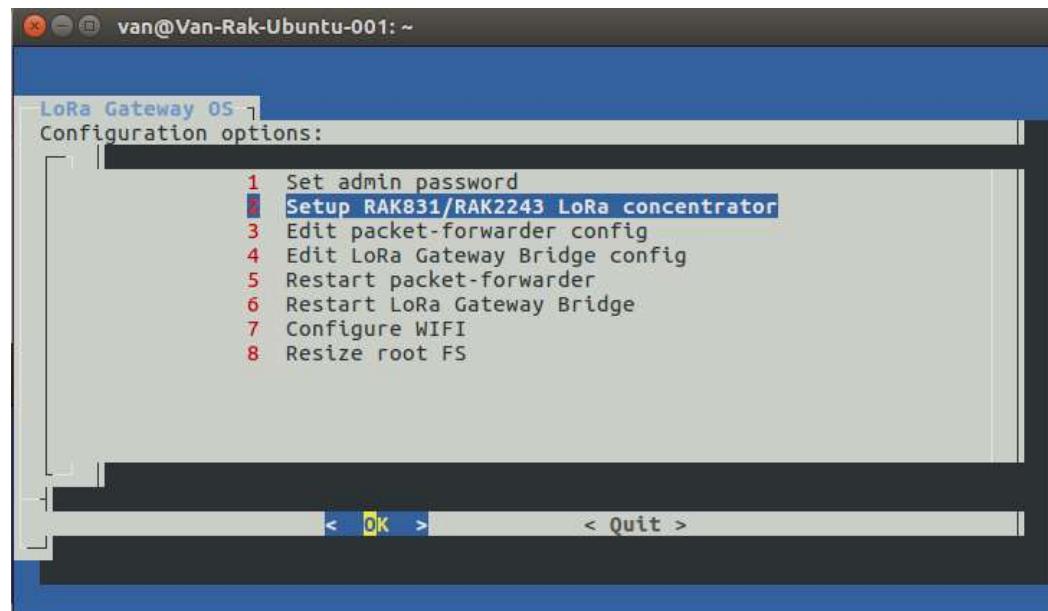
```
jimmy@jimmy-Rak: ~
GNU nano 3.0      /etc/lora-packet-forwarder/global_conf.json

"gateway_conf": {
    "gateway_ID": "b827ebffffee8a9f8",
    /* change with default server address/ports, or overwrite in los
    "server_address": "router.eu.thethings.network",
    "serv_port_up": 1700,
    "serv_port_down": 1700,
    /* adjust the following parameters for your network */
    "keepalive_interval": 10,
    "stat_interval": 30,
    "push_timeout_ms": 100,
    /* forward only valid packets */
    "forward_crc_valid": true,
    "forward_crc_error": false,
    "forward_crc_disabled": false,
    /* gps enable */
    "gps": true,
    "gps_tty_path": "/dev/i2c-1",
    "fake_gps": false,
    "ref_latitude": 10,
}

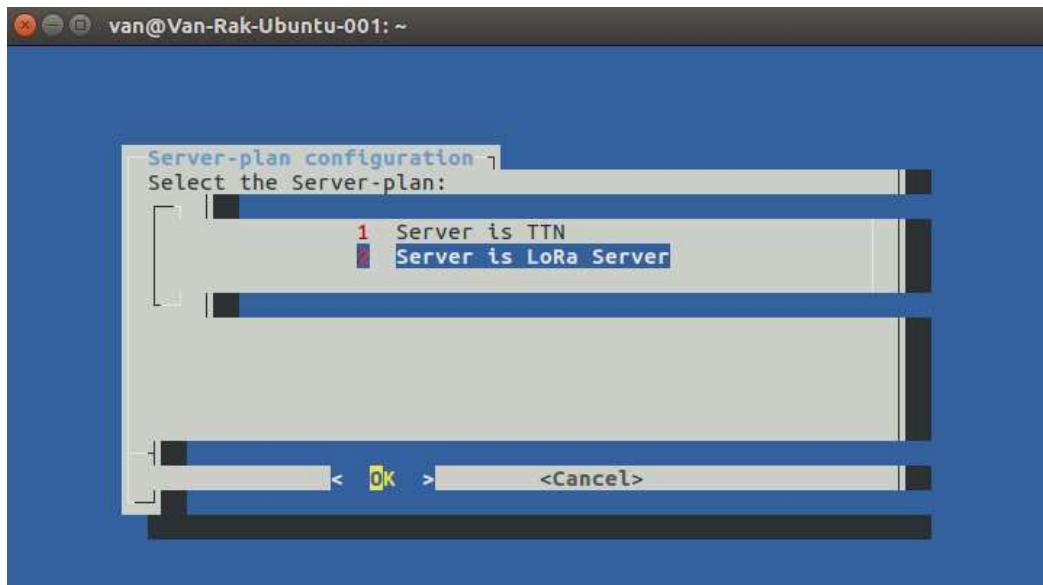
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace  ^U Uncut Text^T To Spell  ^ Go To Line
```

- 8) Now, let's modify the IP address of the LoRa server which the gateway points to, just do as follow:

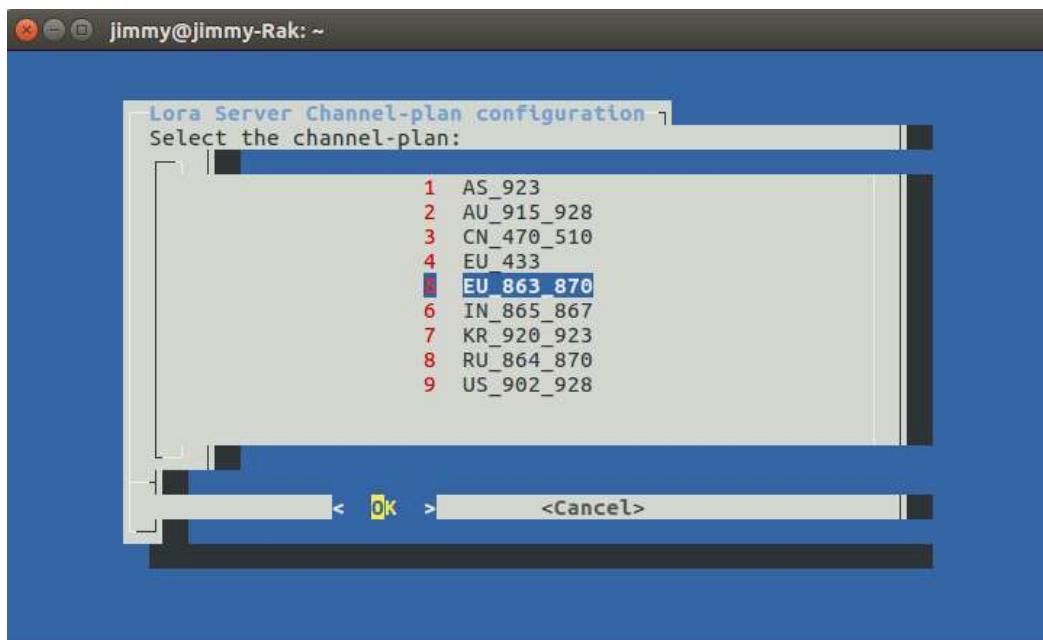
Firstly, select the option 2 “Setup RAK831/RAK2243 LoRa concentrator”.



Then, select the Server-plan 2 “Server is LoRa Server”:



Select the channel-plan 5 to set the EU\_863\_870 frequency.

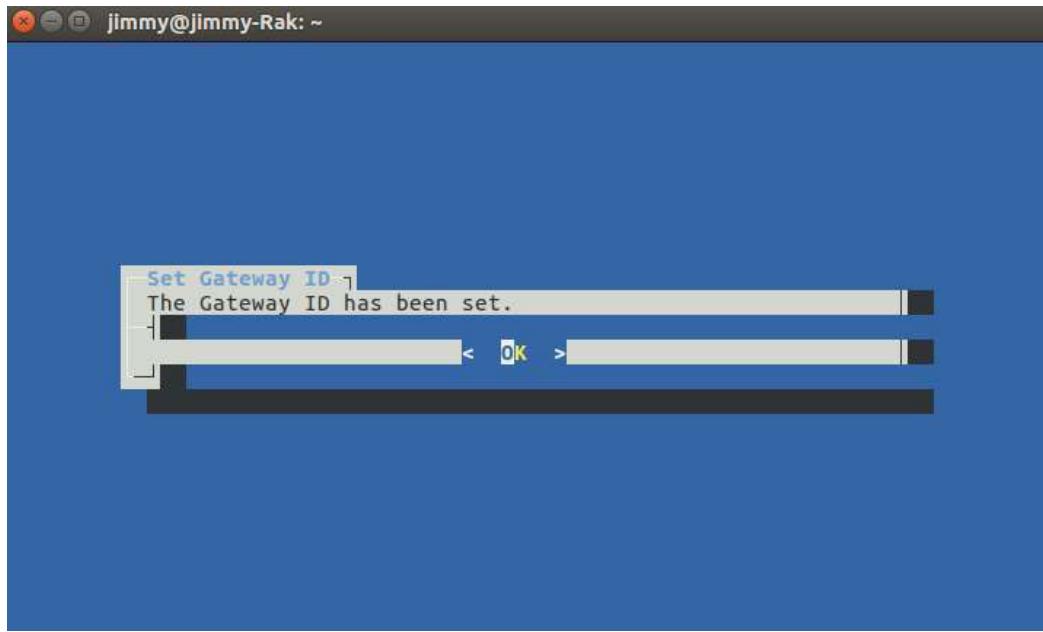


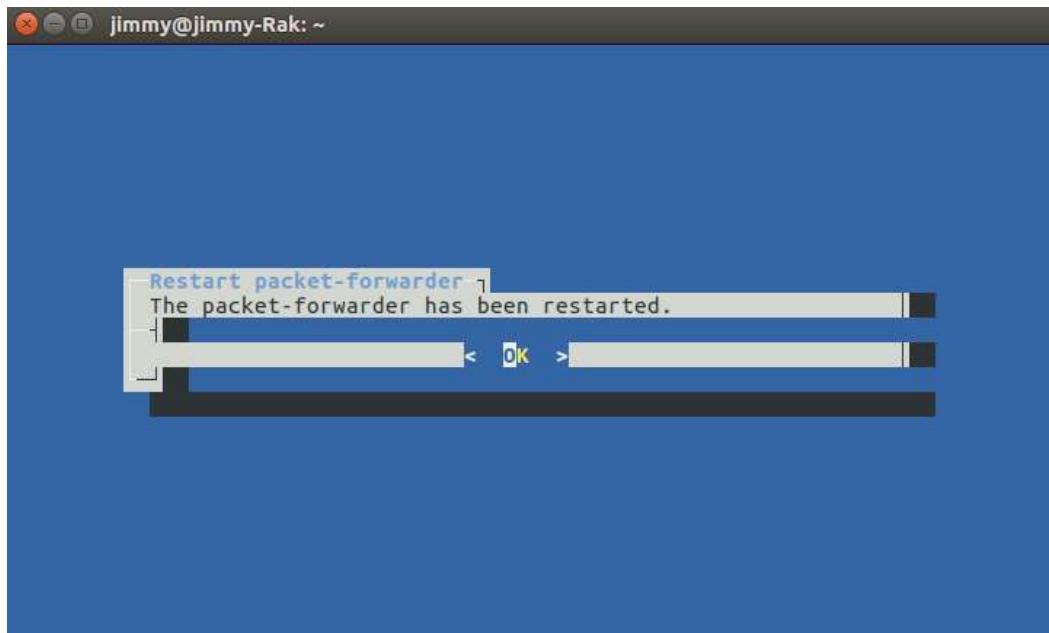
- 9) Enter the IP address of the LoRa server you want to use.



Note: “127.0.0.1” is the default IP address, and it indicates the internal LoRa server. You can enter the IP address of your independent LoRa server which has been setup in the step 1.

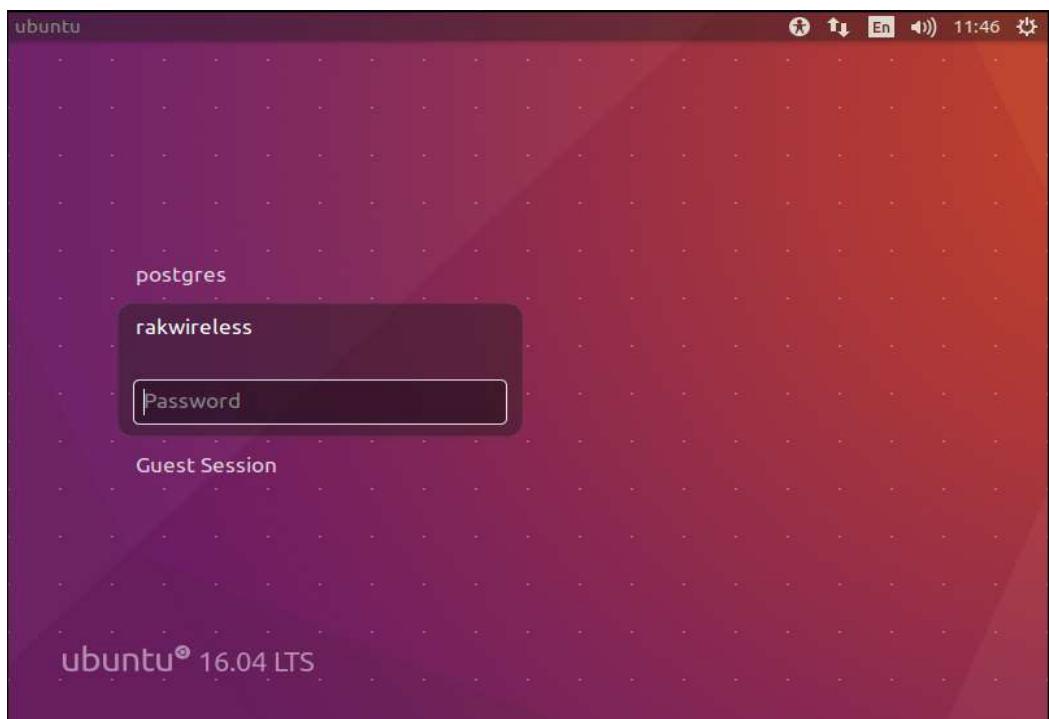
- 10) After enter the IP address of your independent LoRa server and “OK”, you will need two more “OK” as follow:



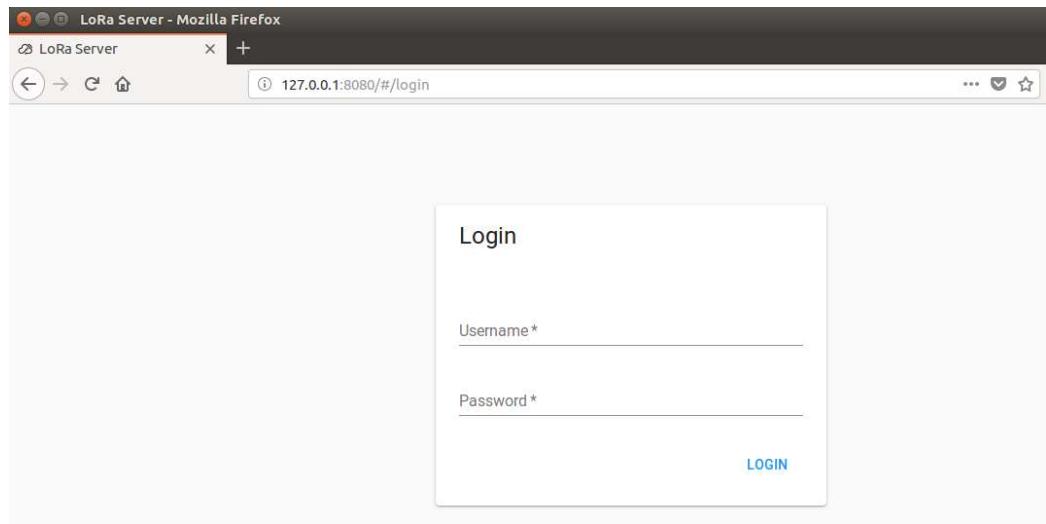


### Step 3: Configure your LoRa server to add a LoRa gateway.

- 1) Login the PC installed the LoRa server OS.



- 2) Open the browser and type <http://127.0.0.1:8080>, you will see the login page of LoRa server as follow:



- 3) Login the LoRa server. The default username and password are both “admin”.

A screenshot of the LoRa Server application interface. The left sidebar has a tree view with nodes: Network-servers, Gateway-profiles, Organizations, All users, and a dropdown menu for 'loraserver' which is expanded to show: Org. settings, Org. users, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups. The main content area is titled "Applications". It features a table with columns: ID, Name, Service-profile, and Description. There is a search bar at the top right with the placeholder "Search organization, application, gateway or device". A "CREATE" button is located in the top right corner of the table area. At the bottom of the table, there is a pagination control with "Rows per page: 10", "0-0 of 0", and navigation arrows.

- 4) Click the “Gateways” in the left menu of the page. There is already a LoRa gateway which has been added, but it is only used to show as a demo and you can also delete it.

The screenshot shows the LoRaServer web interface. The left sidebar contains navigation links: Network-servers, Gateway-profiles, Organizations, All users, vanLoraServer (selected), Org. settings, Org. users, Service-profiles, Device-profiles, Gateways (highlighted with a red box), Applications, and Multicast-groups. The main content area is titled 'Gateways' and displays a table with one row. The table columns are Name, Gateway ID, and Gateway activity (30d). The single row shows 'test\_gateway' and 'aabbcdddeeff1122'. Below the table are pagination controls: Rows per page: 10, 1-1 of 1, and navigation arrows.

Name	Gateway ID	Gateway activity (30d)
test_gateway	aabbcdddeeff1122	

- 5) Create a new item for your LoRa gateway by Click the “CREATE” button in the upper right corner of the following page.

This screenshot is identical to the one above, but the 'CREATE' button in the top right corner of the main content area is highlighted with a red box.

- 6) Then, you can see the following page and just fill them as follow except “Gateway name”, “Gateway description”, and “Gateway ID”. “Gateway name” and “Gateway description” can be filled in with any contents you want, but you must fill in “Gateway ID” with the gateway EUI/ID which you

got in the step 2.

Gateways / Create

Gateway name \*

test\_gateway

The name may only contain words, numbers and dashes.

Gateway description \*

test\_gateway

Gateway ID \*

AABBCCDDEEFF1122

Network-server \*

test\_network\_servers

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

An optional gateway-profile which can be assigned to a gateway. This configuration can be used to automatically re-configure the gateway when LoRa Gateway Bridge is configured so that it manages the packet-forwarder configuration.

Gateway discovery enabled

When enabled (and LoRa 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 onboard GPS, this value will be set automatically when the network re-aligns statistics

- 7) If your LoRa gateway has connected with the LoRa server (ping successfully), you can see the following state page.

Gateway details

Gateway ID: 000000000000868  
Altitude: 0 meters  
GPS coordinates: 0, 0  
Last seen: a few seconds ago

Frames received

8) Congratulations! That means your LoRa gateway has been online now!

Type	Time	Message	DevEUI
DOWNLINK	2:20:30 PM	UnconfirmedDataDown	009f289f
UPLINK	2:20:30 PM	ConfirmedDataUp	009f289f
DOWNLINK	2:20:19 PM	JoinAccept	
UPLINK	2:20:19 PM	JoinRequest	100000000000868

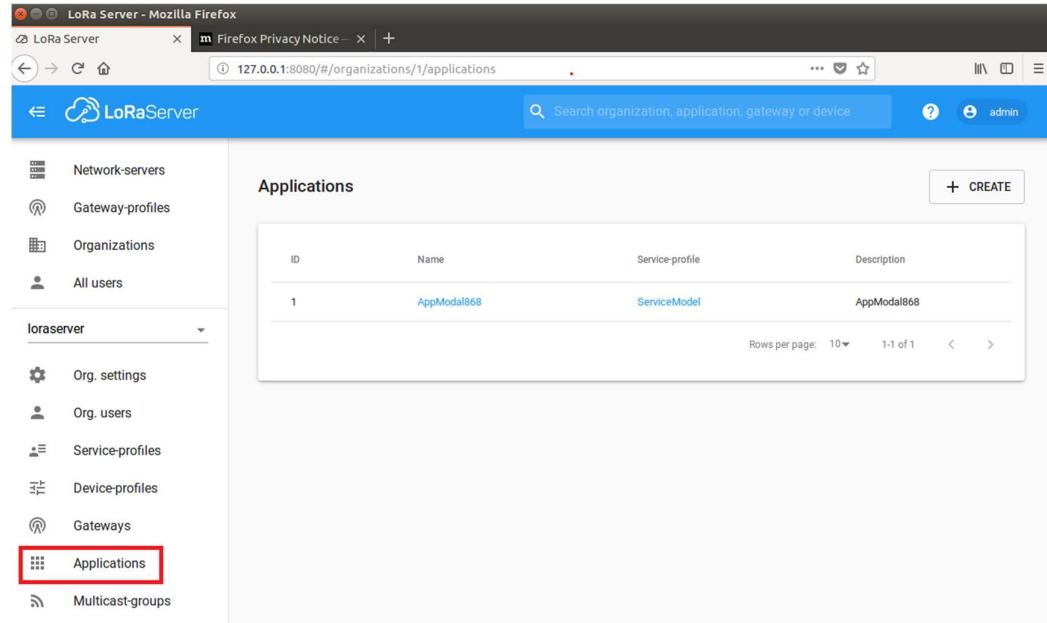
#### Step 4: Configure your LoRa application server to add a LoRa node.

Actually, you can skip this step by using the preconfigured LoRa node content which we have been placed in the LoRa box you bought, including dev\_eui, app\_eui,

and app\_key, just for testing.

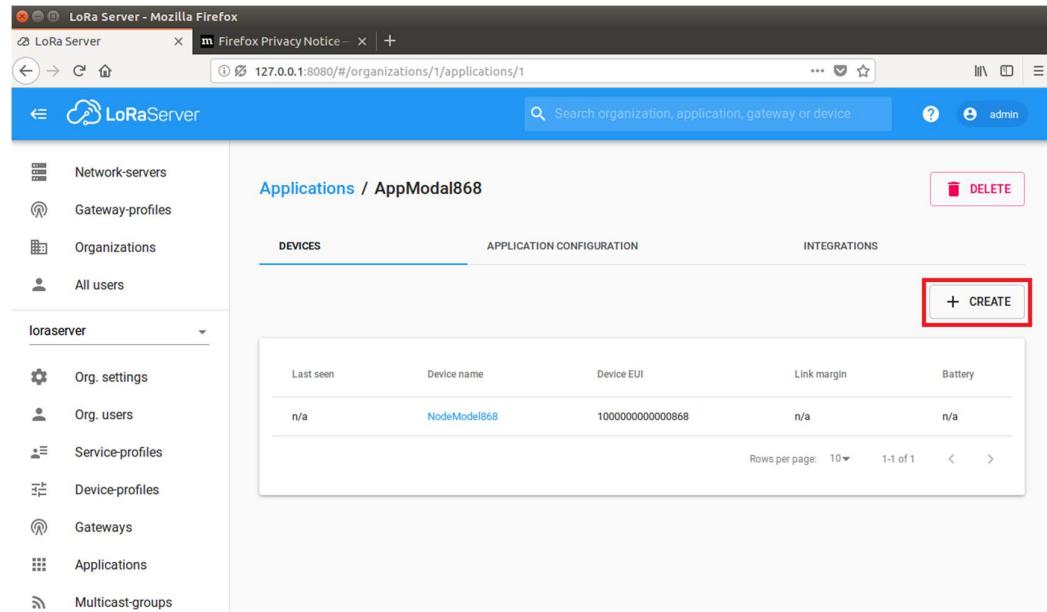
If you want to create a new LoRa node in your application server, please do it as follow:

- 1) Click “Application” in the left menu of the following page.



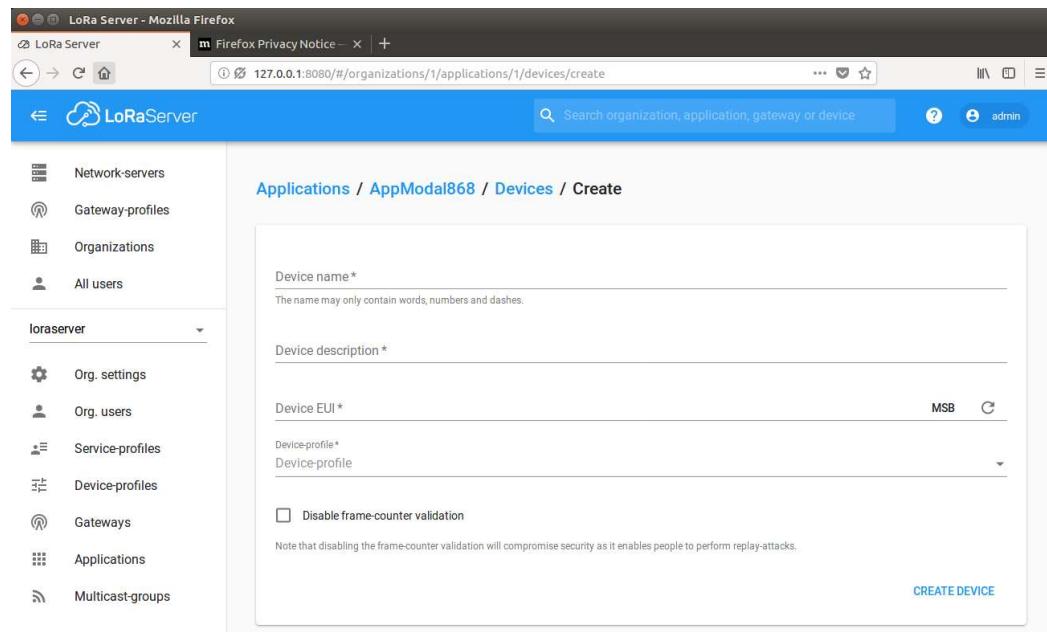
The screenshot shows a browser window for "LoRa Server - Mozilla Firefox". The URL is 127.0.0.1:8080/#/organizations/1/applications. The main content area is titled "Applications" and displays a table with one row. The table columns are ID, Name, Service-profile, and Description. The single entry is ID 1, Name AppModal868, Service-profile ServiceModel, and Description AppModal868. Below the table are pagination controls: "Rows per page: 10" and "1-1 of 1". The left sidebar has a tree structure under "loraserver": Network-servers, Gateway-profiles, Organizations, All users, Applications (which is selected and highlighted with a red box), Multicast-groups. The top right corner shows the user "admin".

- 2) Click the existed application name, like “AppModal868” in the above page, to enter the following page, then click the “CREATE” button to add a LoRa node.



The screenshot shows a browser window for "LoRa Server - Mozilla Firefox" with the URL 127.0.0.1:8080/#/organizations/1/applications/1. The main content area is titled "Applications / AppModal868". It has three tabs: DEVICES (selected), APPLICATION CONFIGURATION, and INTEGRATIONS. In the DEVICES tab, there is a table with one row. The columns are Last seen, Device name, Device EUI, Link margin, and Battery. The row contains n/a, NodeModel868, 1000000000000868, n/a, and n/a. Below the table are pagination controls: "Rows per page: 10" and "1-1 of 1". The left sidebar is identical to the previous screenshot. The top right corner shows the user "admin". A red box highlights the "+ CREATE" button in the top right corner of the main content area.

- 3) Fill in “Device name” and “Device description” freely. Fill in “Device EUI” or click  to generate randomly. Select a item for “Device-profile” and select the check box “Disable frame-counter validation”. Then click “CREATE DEVICE” in the lower right corner.



Device name\*

The name may only contain words, numbers and dashes.

Device description\*

Device EUI\*

Device-profile\*

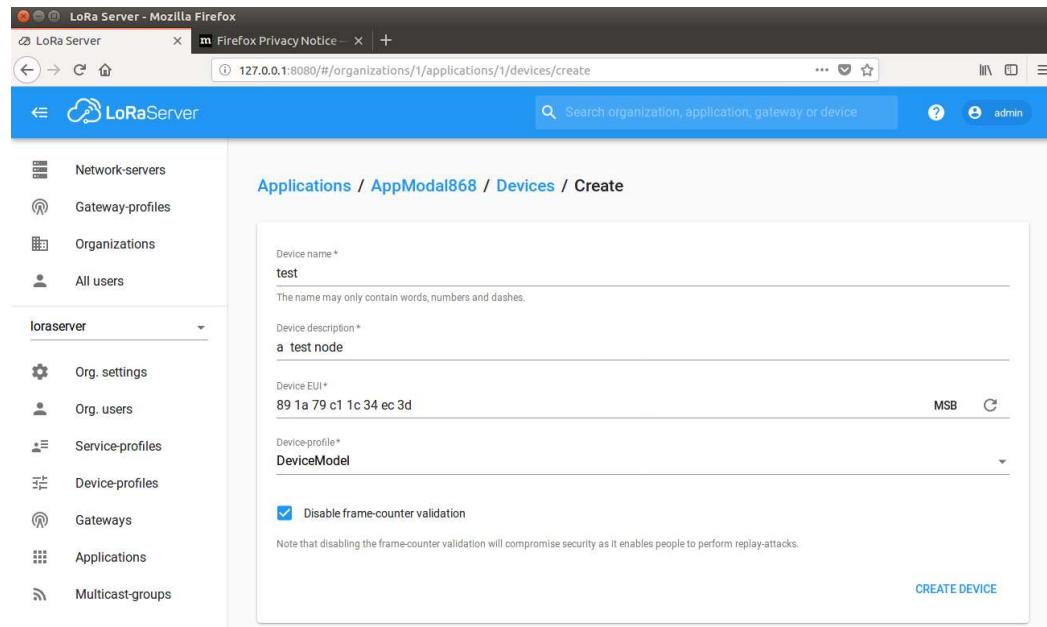
Device-profile

Disable frame-counter validation

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

CREATE DEVICE

For example:



Device name\*

test

The name may only contain words, numbers and dashes.

Device description\*

a test node

Device EUI\*

89 1a 79 c1 1c 34 ec 3d

Device-profile\*

DeviceModel

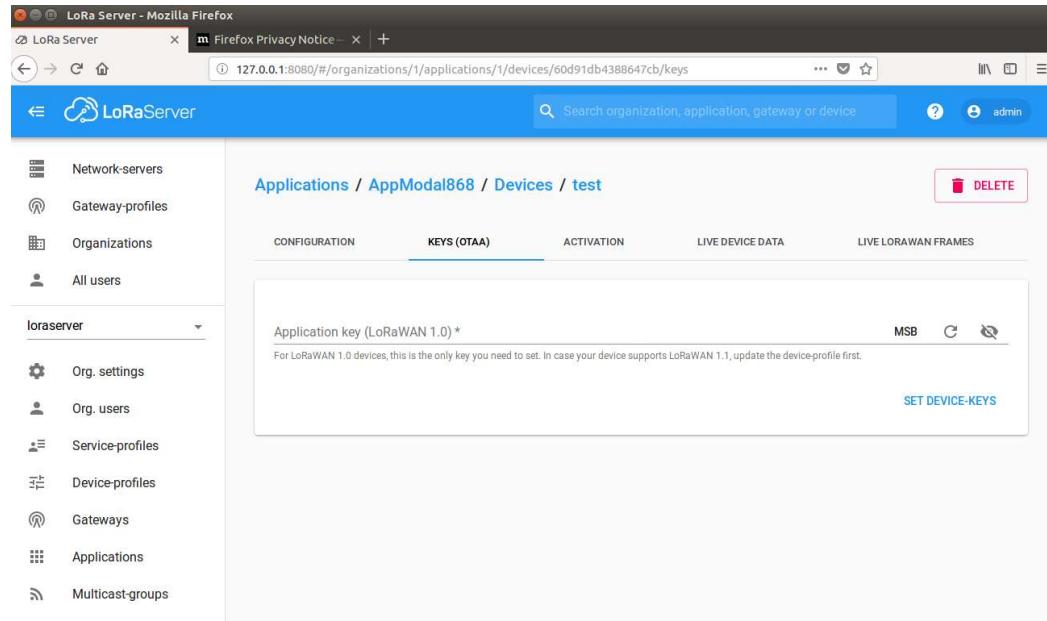
Disable frame-counter validation

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

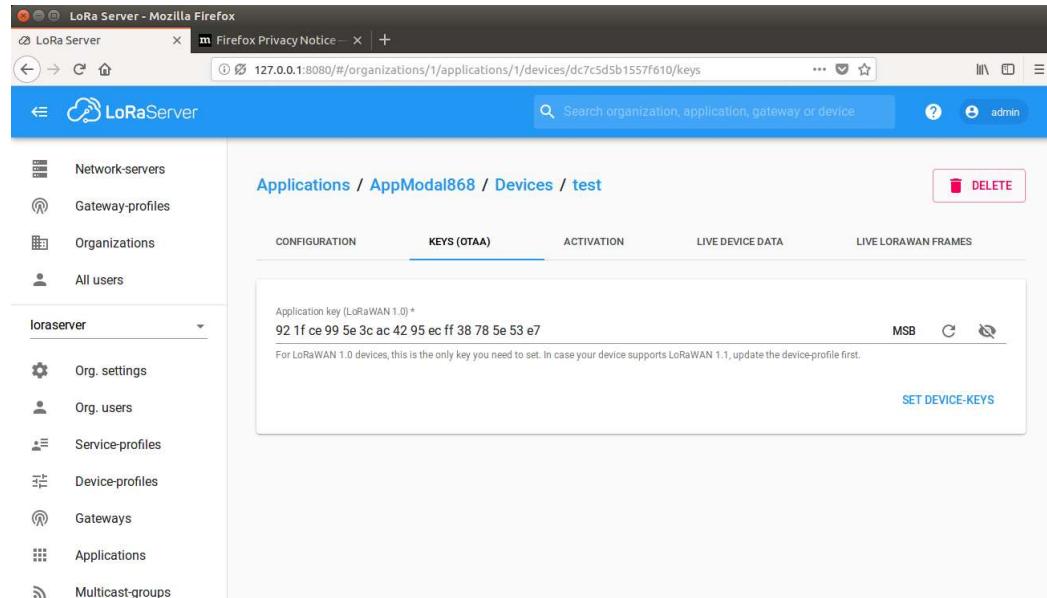
CREATE DEVICE

**Note:** Remember the value of Device EUI you have set, because it will be used on your LoRa node as dev\_eui.

- 4) Set Application key. You can also click  to generate randomly. Then click "SET DEVICE-KEYS".



The screenshot shows the LoRaServer application configuration interface. On the left, a sidebar lists organizational structures: Network-servers, Gateway-profiles, Organizations, and All users. Under 'loraserver', it shows Org. settings, Org. users, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups. The main content area is titled 'Applications / AppModal868 / Devices / test'. It has tabs for Configuration, Keys (OTAA), Activation, Live Device Data, and Live Lorawan Frames. The 'Keys (OTAA)' tab is selected. A text input field contains the placeholder 'Application key (LoRaWAN 1.0)\*'. Below the input field, a note reads: 'For LoRaWAN 1.0 devices, this is the only key you need to set. In case your device supports LoRaWAN 1.1, update the device-profile first.' To the right of the input field are buttons for MSB, a copy icon, and a search icon. At the bottom right of the input field is a blue 'SET DEVICE-KEYS' button.



This screenshot shows the same application configuration interface after a key has been set. The 'Keys (OTAA)' tab is still selected. The input field now contains the generated key: '92 1f ce 99 5e 3c ac 42 95 ec ff 38 78 5e 53 e7'. The note below the input field remains the same. The 'SET DEVICE-KEYS' button is visible at the bottom right of the input field.

Note: Remember the value of this key you have set, because it will be used on your LoRa node as app\_key.

- 5) OK, you've added a LoRa node successfully now.

Last seen	Device name	Device EUI	Link margin	Battery
n/a	NodeModel868	1000000000000868	n/a	n/a
n/a	test	60d91db4388647cb	n/a	n/a

- 6) Once a LoRa node, which has been set with the dev\_eui and app\_key generated above, joins and sends data, you will see the contents on the following pages.

CONFIGURATION	KEYS (OTAA)	ACTIVATION	LIVE DEVICE DATA	LIVE LORAWAN FRAMES
This device has not (yet) been activated.				

What? There is no existed Applications item like the following page showing or you want to add a new Applications item? Just do something as follow to create a new application before creating a new LoRa node. Otherwise, just skip to the step 5.

The screenshot shows the LoRa Server application management interface in Mozilla Firefox. The URL is 127.0.0.1:8080/#/organizations/1/applications. The left sidebar shows navigation options like Network-servers, Gateway-profiles, Organizations, All users, and a dropdown for 'loraserver' which includes Org. settings, Org. users, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups. The main content area is titled 'Applications' and contains a table with columns ID, Name, Service-profile, and Description. A search bar at the top right says 'Search organization, application, gateway or device'. A 'CREATE' button is located in the top right corner of the main content area.

Click “CREATE” to add a new application. You can see the following page and fill in “Application name” and “Application description” with any contents freely, select an item for “Service-profile”. Then you need to select the “Custom JavaScript codec functions” item for “Payload Codec” to decode the received uplink payloads from bytes to a meaningful data object and to encode downlink data objects to bytes.

The screenshot shows the 'Applications / Create' form in Mozilla Firefox. The URL is 127.0.0.1:8080/#/organizations/1/applications/create. The left sidebar is identical to the previous screenshot. The main form has fields for 'Application name\*' (with a note: 'The name may only contain words, numbers and dashes.'), 'Application description \*', 'Service-profile \*' (with a note: 'Select service-profile. The service-profile to which this application will be attached. Note that you can't change this value after the application has been created.'), and 'Payload codec' (with a note: 'By defining a payload codec, LoRa App Server can encode and decode the binary device payload for you.'). At the bottom right of the form is a 'CREATE APPLICATION' button.

For example:

The screenshot shows the LoRa Server application interface in Mozilla Firefox. The URL is 127.0.0.1:8080/#/organizations/1/applications/1/edit. The left sidebar shows 'Network-servers', 'Gateway-profiles', 'Organizations' (selected), and 'All users'. A dropdown for 'loraserver' is open. The main content area is titled 'Applications / AppModal868'. It has tabs for 'DEVICES', 'APPLICATION CONFIGURATION' (selected), and 'INTEGRATIONS'. The 'APPLICATION CONFIGURATION' tab contains fields for 'Application name\*' (AppModal868) and 'Application description\*' (AppModal868). Below these is a section for 'Payload codec' with 'Custom JavaScript codec functions'. The code editor contains the following JavaScript code:

```

1 // Decode decodes an array of bytes into an object.
2 // - fPort contains the LoRaWAN fPort number
3 // - bytes is an array of bytes, e.g. [225, 230, 255, 0]
4 // - The function must return an object, e.g. {"temperature": 22.5}
5 function bin2String(array) {
6   return String.fromCharCode.apply(String, array);
7 }
8
9 function bin2HexStr(arr)
10 {
11   var str = "";
12   for(var i=0; i<arr.length; i++)
13   {
14     if (i != 0)
15     {
16       str += ",";
17     }
18     var tmp = arr[i].toString(16);
19     if(tmp.length == 1)
20       {
21
22
23
24
25
26
27
28
29
2

```

The note below the code states: 'The function must have the signature function Decode(fPort, bytes) and must return an object. LoRa App Server will convert this object to JSON.'

Note: When you select “Custom JavaScript codec functions” item for “Payload Codec”, you need to copy the following contents to the first source code edit box on the above page:

```

// 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]

// The function must return an object, e.g. {"temperature": 22.5}

function bin2String(array)

{

    return String.fromCharCode.apply(String, array);

}

function bin2HexStr(arr)

{



var str = "";
```

```

        for(var i=0; i<arr.length; i++)
        {
            if (i != 0)
            {
                str += ",";
            }

            var tmp = arr[i].toString(16);

            if(tmp.length == 1)
            {
                tmp = "0" + tmp;
            }

            str += "0x";
            str += tmp;
        }

        return str;
    }

    function Decode(fPort, bytes)
    {
        var myObj = {"DecodeDataString": "", "DecodeDataHex": ""};

        var tostring=bin2String(bytes);

        var toshexstring=bin2HexStr(bytes);

        myObj.DecodeDataString = tostring;

        myObj.DecodeDataHex = toshexstring;

        return myObj;
    }
}

```

Then copy the following contents to the second source code edit box:

```

// 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}

// The function must return an array of bytes, e.g. [225, 230, 255, 0]

function Encode(fPort, obj)

{
    return [];
}

```

### Step 5: Set your LoRa node to join the LoRaWAN network.

In this tutorial, we use RAK811 as LoRa node to show how a LoRa node could join the LoRaWAN network.

- 4) Set the dev\_eui and app\_key, which you got in the step 4, into the LoRa node through AT command.

For example:

```

[10:03:47.639]发->◇at+set_config=dev_eui:10000000000000868
□
[10:03:48.808]收-<◆OK

[10:03:59.414]发->◇at+set_config=app_eui:70B3D57ED00157D1
□
[10:04:00.584]收-<◆OK

[10:04:12.070]发->◇at+set_config=app_key:1111111111111111111111111111111868
□
[10:04:13.241]收-<◆OK
|
```

- 5) Join the LoRaWAN network through AT command.

“at+recv=3,0,0” indicates that the LoRa node has joined the LoRaWAN network successfully.

```

[10:49:19.615]发->◇at+join=otaa
□
[10:49:19.657]收-<◆OK
[10:49:26.159]收-<◆at+recv=3,0,0
|
```

- 6) Try to send some data to the application server.

For example:

```

[10:50:33.991]发->◇at+send=1,3,48454c4c4f20574f524c44
□
[10:50:34.002]收-<◆OK
[10:50:37.837]收-<◆at+recv=1,0,0
|
```

“at+recv=1,0,0” indicates the LoRa node has sent the data successfully.

- 7) Now you can see the contents sent by LoRa node on the LoRa application page as follow.

Applications / AppModal868 / Devices / NodeModel868

ACTION	KEYS (OTAA)	ACTIVATION	LIVE DEVICE DATA	LIVE LORAWAN FRAMES
DOWNLINK	9:45:20 AM	JoinAccept		
UPLINK	9:45:20 AM	JoinRequest	1000000000000868	

Applications / AppModal868 / Devices / NodeModel868

ACTION	KEYS (OTAA)	ACTIVATION	LIVE DEVICE DATA	LIVE LORAWAN FRAMES

9:54:34 AM      uplink

```

adr: true
applicationID: "1"
applicationName: "AppModal868"
data: "SEVMT8bgV09STEQ="
devEUI: "1000000000000868"
deviceName: "NodeModel868"
fCnt: 2
fPort: 3
object: { 2 keys
  DecodeDataHex: "0x48,0x45,0x4c,0x4c,0x4f,0x20,0x57,0x4f,0x52,0x4c,0x44"
  DecodeDataString: "HELLO WORLD"
}
  0: { 5 keys
    gatewayID: "0000000000000868"
    LoRaSNR: 8.5
    location: { 3 keys
      altitude: 0
    }
  }
}

```

## **Case 3: I want to set up a LoRa network quickly, but I want to use TTN as my LoRa server.**

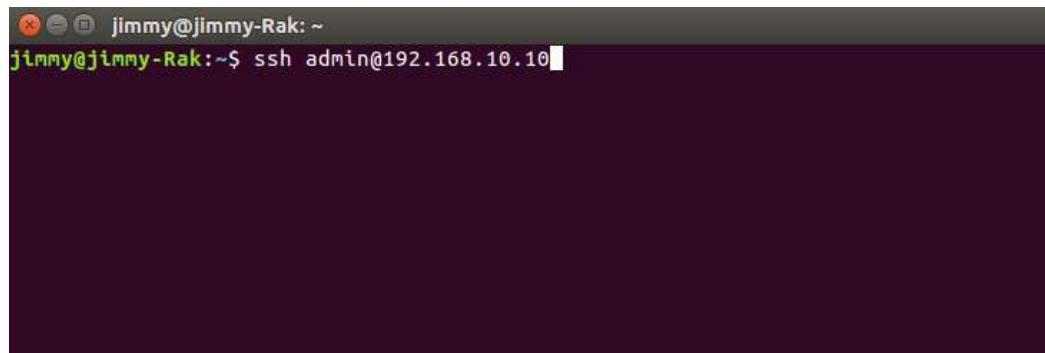
Just do as the following steps:

### **Step 1: Configure your LoRa gateway**

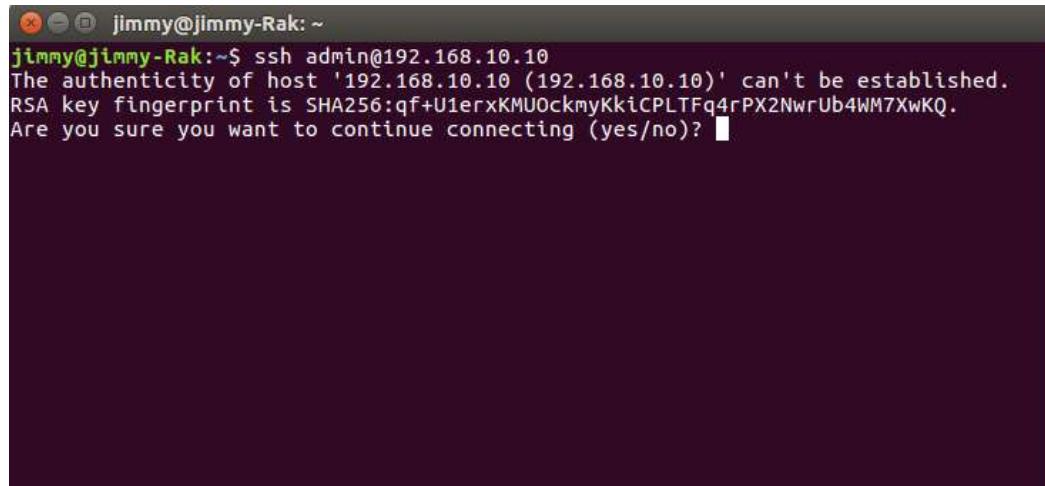
- 1) Login the Raspberry Pi through SSH, the default username and password are both “admin”.

**Note:** By default, the IP address of the LoRa gateway ethernet is 192.168.10.10, and you can connect to it through wire using a PC and SSH.

For example:



```
jimmy@jimmy-Rak: ~
jimmy@jimmy-Rak:~$ ssh admin@192.168.10.10
```



```
jimmy@jimmy-Rak: ~
jimmy@jimmy-Rak:~$ ssh admin@192.168.10.10
The authenticity of host '192.168.10.10 (192.168.10.10)' can't be established.
RSA key fingerprint is SHA256:qf+U1erxKMU0ckmyKkiCPLTFq4rPX2NwrUb4WM7XwKQ.
Are you sure you want to continue connecting (yes/no)?
```

```
jimmy@jimmy-Rak:~$ ssh admin@192.168.10.10
The authenticity of host '192.168.10.10 (192.168.10.10)' can't be established.
RSA key fingerprint is SHA256:qf+U1erxKMU0ckmyKkiCPLTFq4rPX2NwrUb4WM7XwKQ.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/jimmy/.ssh/known_hosts)
.
admin@192.168.10.10's password: █
```

**Note:** If you don't know how to use SSH, please google or Baidu.

- 2) After that, you will see the following page, which indicates you have login successfully:

- 3) Enter the command “`sudo gateway-config`” and the login password to configure the gateway. This command has been shown in the page:

```
jimmy@jimmy-Rak:~$ ssh admin@192.168.10.10
The authenticity of host '192.168.10.10 (192.168.10.10)' can't be established.
RSA key fingerprint is SHA256:qf+U1erxKMU0ckmyKkiCPLTFq4rPX2NwrUb4WM7XwKQ.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/jimmy/.ssh/known_hosts)
.
admin@192.168.10.10's password:

documentation and copyright information: www.loraserver.io

Commands:
> sudo gateway-config - configure the gateway
> sudo monit status - display service monitor

raspberrypi3:~$
```

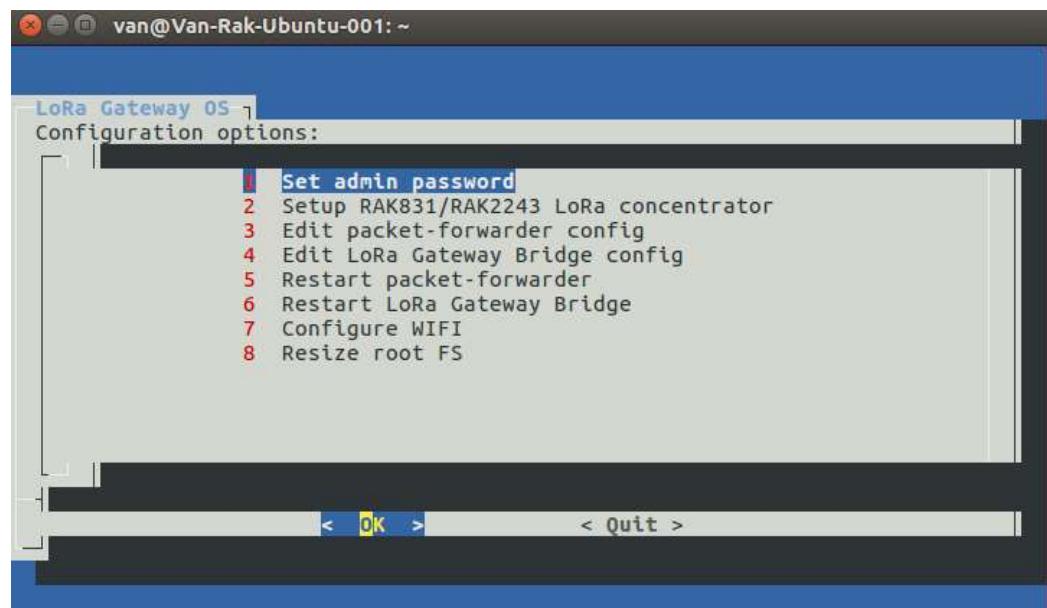
```
jimmy@jimmy-Rak:~$ ssh admin@192.168.10.10
The authenticity of host '192.168.10.10 (192.168.10.10)' can't be established.
RSA key fingerprint is SHA256:qf+U1erxKMU0ckmyKkiCPLTFq4rPX2NwrUb4WM7XwKQ.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/jimmy/.ssh/known_hosts)
.
admin@192.168.10.10's password:

documentation and copyright information: www.loraserver.io

Commands:
> sudo gateway-config - configure the gateway
> sudo monit status - display service monitor

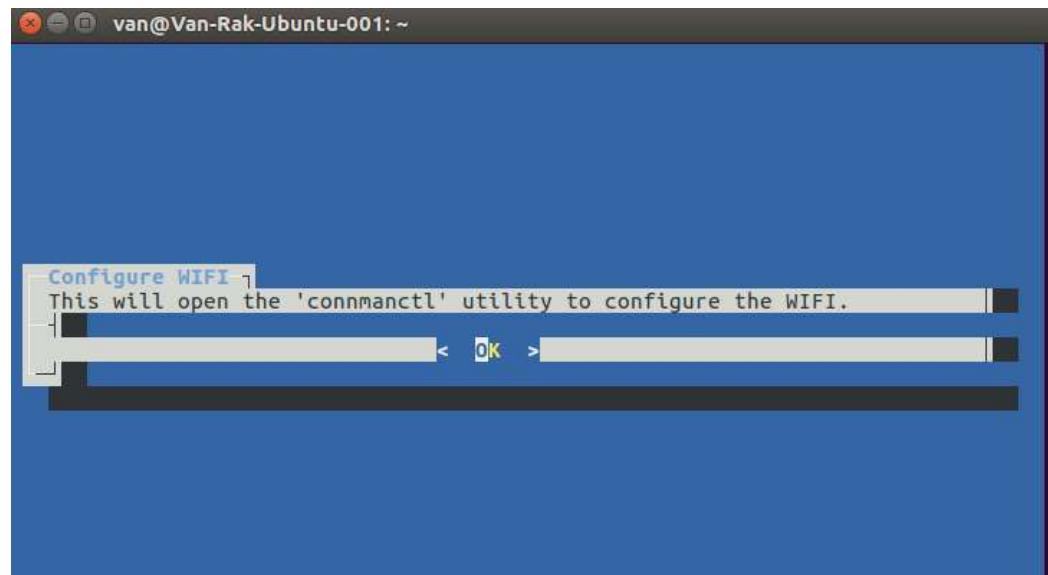
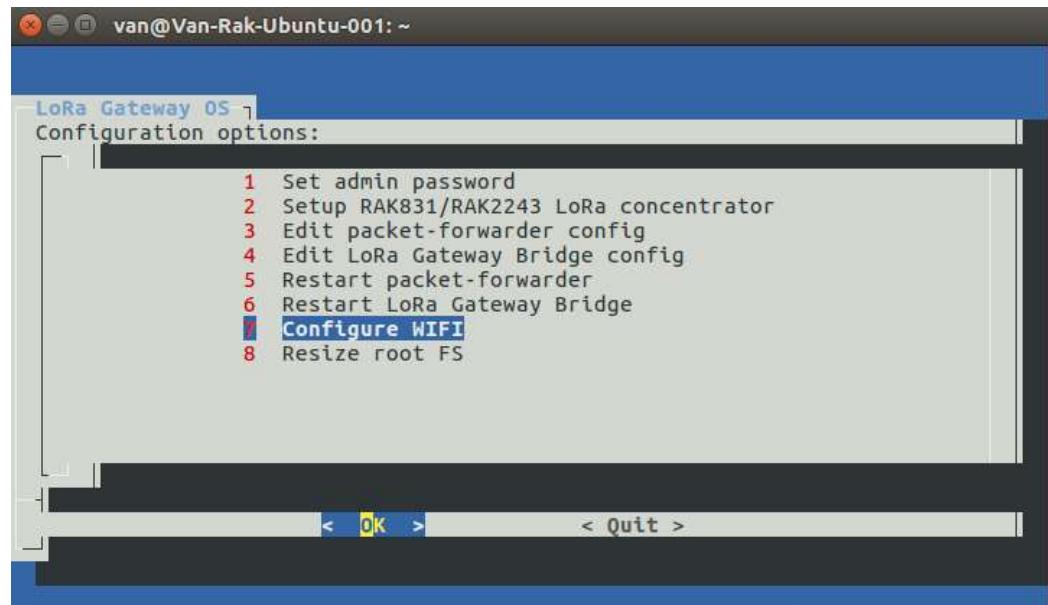
raspberrypi3:~$ sudo gateway-config
```

- 4) Now, you should see a configuration page as follow:



**Note:** In this tutorial, you may only use the option 1, 2, 3, and 7.

- 5) If you want to connect the LoRa gateway to a WiFi network, just do as follow:



```
van@Van-Rak-Ubuntu-001: ~
connmanctl quickstart
1) Enable wifi:
enable wifi

2) Scan available wifi networks:
scan wifi

3) Display available wifi networks:
services

4) Turn on agent:
agent on

5) Connect to network:
connect wifi_...

6) Quit connmanctl:
quit

[ OK ]
```

```
van@Van-Rak-Ubuntu-001: ~
Error getting VPN connections: The name net.connman.vpn was not provided by any
connmanctl> [ ]
```

Enable WiFi:

```
van@Van-Rak-Ubuntu-001: ~
Error getting VPN connections: The name net.connman.vpn was not provided by any
connmanctl> enable wifi
Enabled wifi
connmanctl> [ ]
```

Scan WiFi:

```
van@Van-Rak-Ubuntu-001: ~
Error getting VPN connections: The name net.connman.vpn was not provided by any
connmanctl> enable wifi
Enabled wifi
connmanctl> scan wifi
Scan completed for wifi
connmanctl>
```

Check services:

```
van@Van-Rak-Ubuntu-001: ~
Error getting VPN connections: The name net.connman.vpn was not provided by any
connmanctl> enable wifi
Enabled wifi
connmanctl> scan wifi
Scan completed for wifi
connmanctl> services
*AR Wired          ethernet_b827ebe8a9f8_cable
                  wifi_b827ebbdxfcad_4163636573734d65_managed_psk
AccessMe           wifi_b827ebbdxfcad_4163636573734d6554657374_managed_psk
AccessMeTest       wifi_b827ebbdxfcad_53454e44434f4d_managed_psk
SENDCOM            wifi_b827ebbdxfcad_53454e44434f4d_managed_psk
WENSEN             wifi_b827ebbdxfcad_57454e53454e_managed_psk
SENFA              wifi_b827ebbdxfcad_53454e4641_managed_psk
sxxxnyjk          wifi_b827ebbdxfcad_737878786e796a6b_managed_psk
HWJD               wifi_b827ebbdxfcad_48574a44_managed_psk
SXRKKG             wifi_b827ebbdxfcad_5358524b474b_managed_psk
XAYNKJ             wifi_b827ebbdxfcad_5841594e4b4a_managed_psk
HT-WIFI            wifi_b827ebbdxfcad_48542d57494649_managed_psk
connmanctl>
```

Agent on:

```
van@Van-Rak-Ubuntu-001: ~
Error getting VPN connections: The name net.connman.vpn was not provided by any
connmanctl> enable wifi
Enabled wifi
connmanctl> scan wifi
Scan completed for wifi
connmanctl> services
*AR Wired          ethernet_b827ebe8a9f8_cable
                  AccessMe          wifi_b827ebbdxfcad_4163636573734d65_managed_psk
                  AccessMeTest        wifi_b827ebbdxfcad_4163636573734d6554657374_managed_psk
                  SENDCOM           wifi_b827ebbdxfcad_53454e44434f4d_managed_psk
                  SENFA              wifi_b827ebbdxfcad_53454e4641_managed_psk
                  WENSEN             wifi_b827ebbdxfcad_57454e53454e_managed_psk
                  SXRKGK            wifi_b827ebbdxfcad_5358524b474b_managed_psk
                  HWJD               wifi_b827ebbdxfcad_48574a44_managed_psk
                  TAINENG            wifi_b827ebbdxfcad_5441494e454e47_managed_psk
                  XAYNKJ            wifi_b827ebbdxfcad_5841594e4b4a_managed_psk
                  HT-WIFI            wifi_b827ebbdxfcad_48542d57494649_managed_psk
connmanctl> agent on
Agent registered
connmanctl>
```

Connect WiFi as follow (notice what you need to enter isn't the SSID):

```
van@Van-Rak-Ubuntu-001: ~
Error getting VPN connections: The name net.connman.vpn was not provided by any
connmanctl> enable wifi
Enabled wifi
connmanctl> scan wifi
Scan completed for wifi
connmanctl> services
*AR Wired          ethernet_b827ebe8a9f8_cable
                  AccessMe          wifi_b827ebbdxfcad_4163636573734d65_managed_psk
                  AccessMeTest        wifi_b827ebbdxfcad_4163636573734d6554657374_managed_psk
                  SENDCOM           wifi_b827ebbdxfcad_53454e44434f4d_managed_psk
                  SENFA              wifi_b827ebbdxfcad_53454e4641_managed_psk
                  WENSEN             wifi_b827ebbdxfcad_57454e53454e_managed_psk
                  SXRKGK            wifi_b827ebbdxfcad_5358524b474b_managed_psk
                  HWJD               wifi_b827ebbdxfcad_48574a44_managed_psk
                  TAINENG            wifi_b827ebbdxfcad_5441494e454e47_managed_psk
                  XAYNKJ            wifi_b827ebbdxfcad_5841594e4b4a_managed_psk
                  HT-WIFI            wifi_b827ebbdxfcad_48542d57494649_managed_psk
connmanctl> agent on
Agent registered
connmanctl> connect wifi_b827ebbdxfcad_4163636573734d65_managed_psk
Agent RequestInput wifi_b827ebbdxfcad_4163636573734d65_managed_psk
  Passphrase = [ Type=psk, Requirement=mandatory ]
Passphrase? ■
```

After entering the password, if you see “connected ...”, it indicates the LoRa gateway has connected WiFi successfully.

```

van@Van-Rak-Ubuntu-001: ~
Enabled wifi
connmanctl> scan wifi
Scan completed for wifi
connmanctl> services
*AR Wired          ethernet_b827ebe8a9f8_cable
                  wifi_b827ebbdxfcad_4163636573734d65_managed_psk
AccessMe           wifi_b827ebbdxfcad_4163636573734d654657374_managed_psk
AccessMeTest       wifi_b827ebbdxfcad_53454e44434f4d_managed_psk
SENDCOM            wifi_b827ebbdxfcad_53454e4641_managed_psk
SENFA              wifi_b827ebbdxfcad_57454e53454e_managed_psk
WENSEN             wifi_b827ebbdxfcad_737878786e796a6b_managed_psk
Sxxxnyjk          wifi_b827ebbdxfcad_5358524b474b_managed_psk
SXRGK              wifi_b827ebbdxfcad_48574a44_managed_psk
HWJD               wifi_b827ebbdxfcad_48542d57494649_managed_psk
HT-WIFI            wifi_b827ebbdxfcad_5841594e4b4a_managed_psk
XAYNKJ             wifi_b827ebbdxfcad_5441494e454e47_managed_psk
TAINENG            wifi_b827ebbdxfcad_5441494e454e47_managed_psk
connmanctl> agent on
Agent registered
connmanctl> connect wifi_b827ebbdxfcad_4163636573734d65_managed_psk
Agent RequestInput wifi_b827ebbdxfcad_4163636573734d65_managed_psk
  Passphrase = [ Type=psk, Requirement=mandatory ]
Passphrase? rakwireless.2018
Connected wifi_b827ebbdxfcad_4163636573734d65_managed_psk
connmanctl>

```

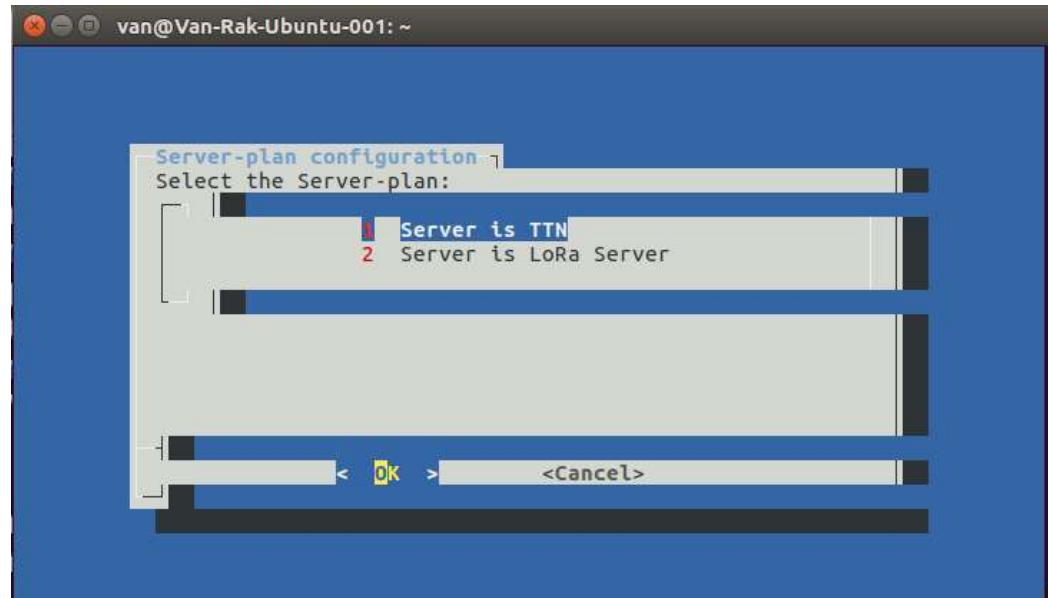
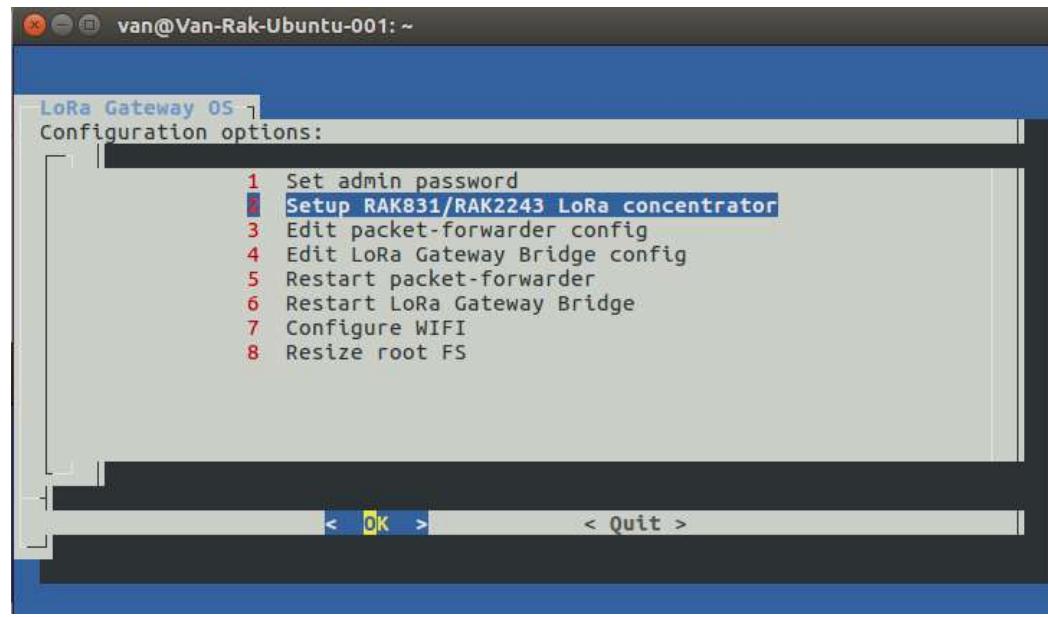
Then quit the WiFi configuration page:

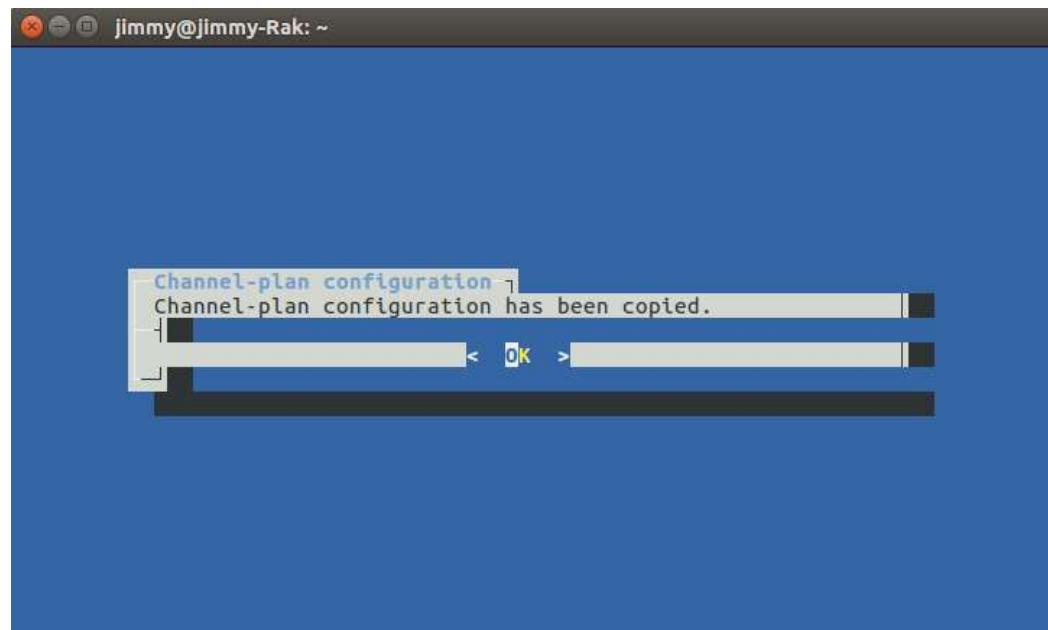
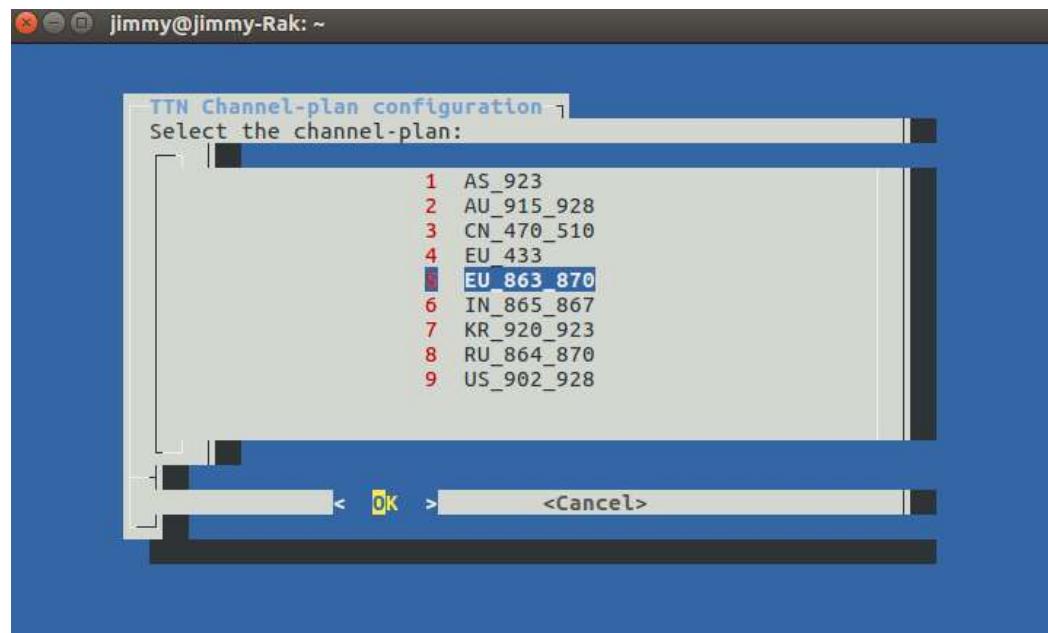
```

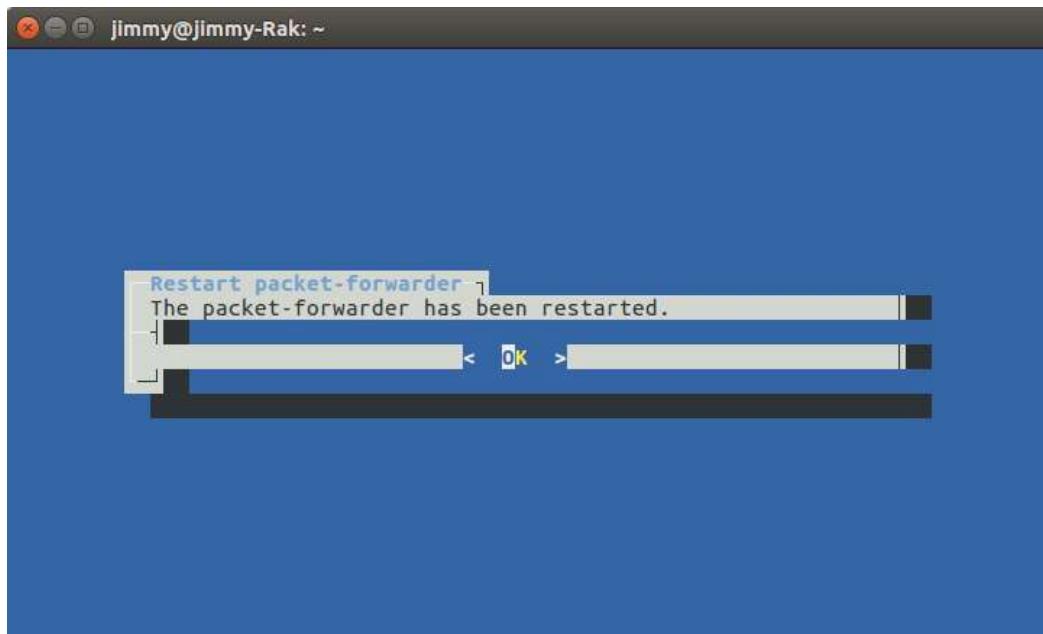
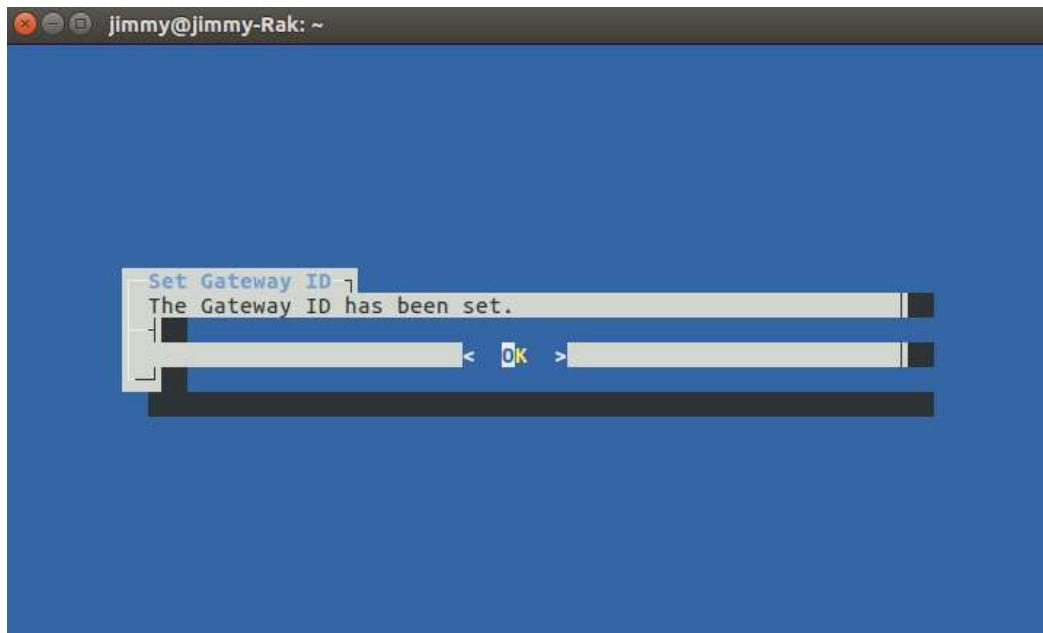
van@Van-Rak-Ubuntu-001: ~
Enabled wifi
connmanctl> scan wifi
Scan completed for wifi
connmanctl> services
*AR Wired          ethernet_b827ebe8a9f8_cable
                  wifi_b827ebbdxfcad_4163636573734d65_managed_psk
AccessMe           wifi_b827ebbdxfcad_4163636573734d654657374_managed_psk
AccessMeTest       wifi_b827ebbdxfcad_53454e44434f4d_managed_psk
SENDCOM            wifi_b827ebbdxfcad_53454e4641_managed_psk
SENFA              wifi_b827ebbdxfcad_57454e53454e_managed_psk
WENSEN             wifi_b827ebbdxfcad_737878786e796a6b_managed_psk
Sxxxnyjk          wifi_b827ebbdxfcad_5358524b474b_managed_psk
SXRGK              wifi_b827ebbdxfcad_48574a44_managed_psk
HWJD               wifi_b827ebbdxfcad_48542d57494649_managed_psk
HT-WIFI            wifi_b827ebbdxfcad_5841594e4b4a_managed_psk
XAYNKJ             wifi_b827ebbdxfcad_5441494e454e47_managed_psk
TAINENG            wifi_b827ebbdxfcad_5441494e454e47_managed_psk
connmanctl> agent on
Agent registered
connmanctl> connect wifi_b827ebbdxfcad_4163636573734d65_managed_psk
Agent RequestInput wifi_b827ebbdxfcad_4163636573734d65_managed_psk
  Passphrase = [ Type=psk, Requirement=mandatory ]
Passphrase? rakwireless.2018
Connected wifi_b827ebbdxfcad_4163636573734d65_managed_psk
connmanctl> quit

```

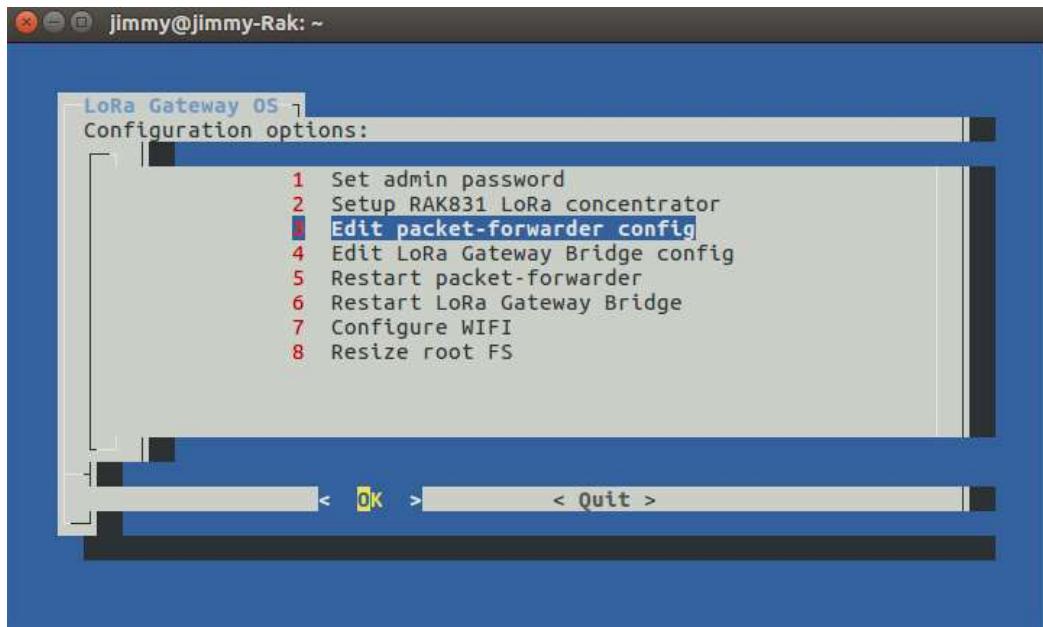
- 6) When you power on this LoRa gateway firstly, it works on EU868(EU\_863\_870) and points to the internal LoRa server by default. Now, let's change the pointed LoRa server to TTN, just do as follow







- 7) OK, that's OK.
- 8) Let's see the global\_conf.json file to check:

A screenshot of a terminal window titled "jimmy@jimmy-Rak: ~". The title bar has standard window controls. The main area shows the contents of the file "/etc/lora-packet-forwarder/global\_conf.json" using the "GNU nano 3.0" editor. The file contains JSON configuration for two radios, "radio\_0" and "radio\_1". The "radio\_0" section includes fields like "enable", "type", "freq", "rss\_offset", "tx\_enable", "tx\_freq\_min", and "tx\_freq\_max". The "radio\_1" section includes similar fields. A status bar at the bottom indicates "[ Read 171 lines ]". A series of keyboard shortcuts are listed at the bottom: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^C Cur Pos, ^X Exit, ^R Read File, ^\ Replace, ^U Uncut Text, ^T To Spell, and ^ Go To Line.

```
jimmy@jimmy-Rak: ~
GNU nano 3.0      /etc/lora-packet-forwarder/global_conf.json

    "gateway_conf": {
        "gateway_ID": "b827ebffffee8a9f8",
        /* change with default server address/ports, or overwrite in los
        "server_address": "router.eu.thethings.network",
        "serv_port_up": 1700,
        "serv_port_down": 1700,
        /* adjust the following parameters for your network */
        "keepalive_interval": 10,
        "stat_interval": 30,
        "push_timeout_ms": 100,
        /* forward only valid packets */
        "forward_crc_valid": true,
        "forward_crc_error": false,
        "forward_crc_disabled": false,
        /* gps enable */
        "gps": true,
        "gps_tty_path": "/dev/i2c-1",
        "fake_gps": false,
        "ref_latitude": 10,
    }

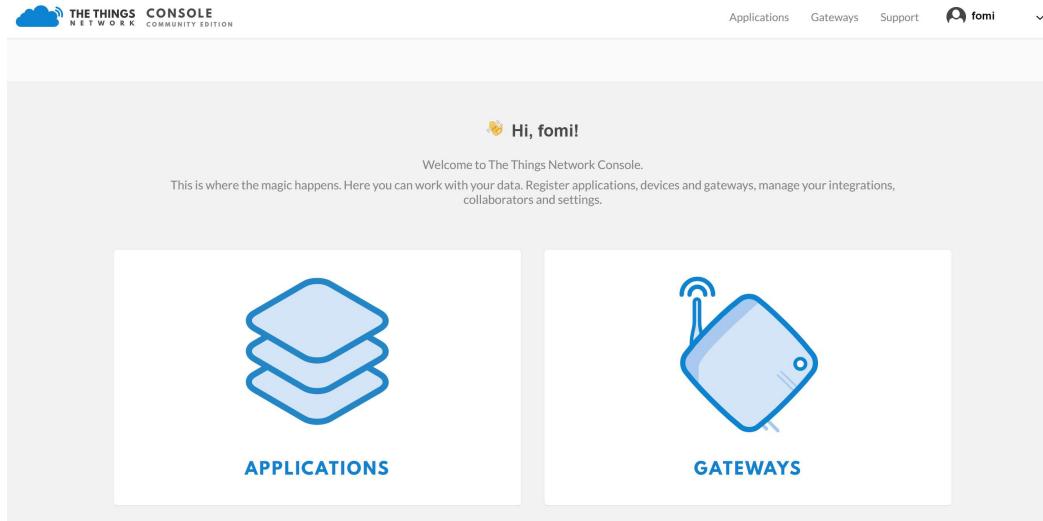
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text^T To Spell  ^ Go To Line
```

**Note:** Do you see the gateway ID in the end of this file as shown as the following picture? It is important to remember it!

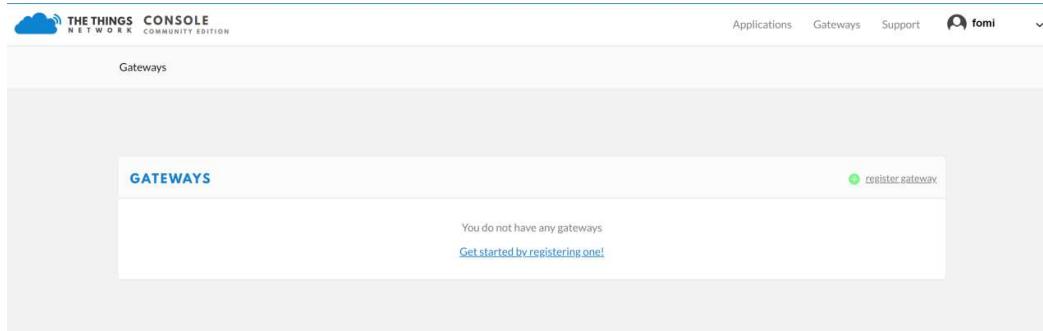
## Step 2: Register your LoRa gateway to TTN LoRa server.

- 1) Login to <https://www.thethingsnetwork.org/>, and open your Console panel;

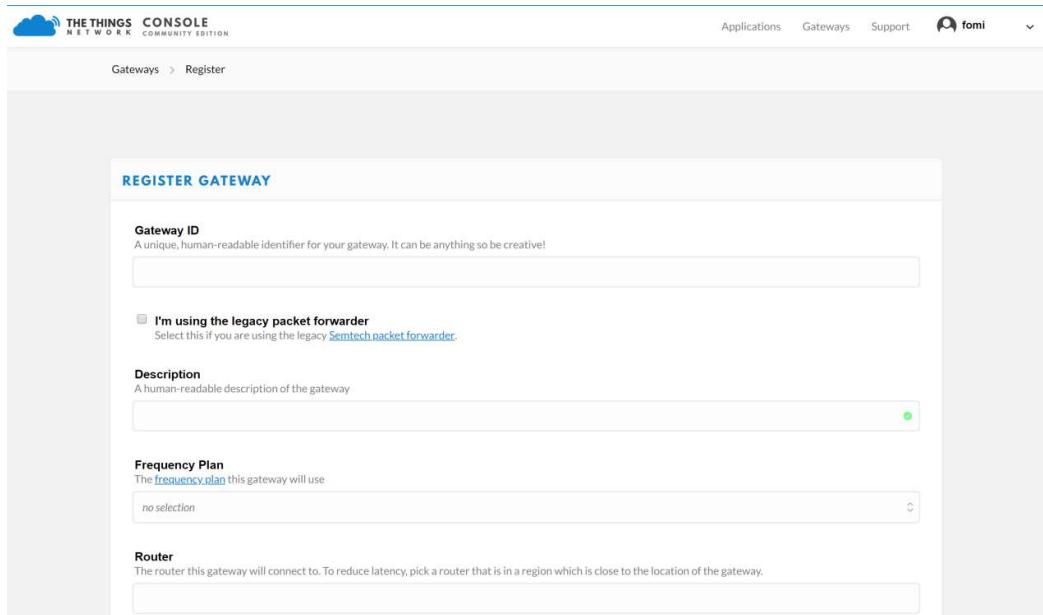




2) Click on “GATEWAYS”, and “register gateway”:



3) You'll see the following page:



4) Fill in them as follow:

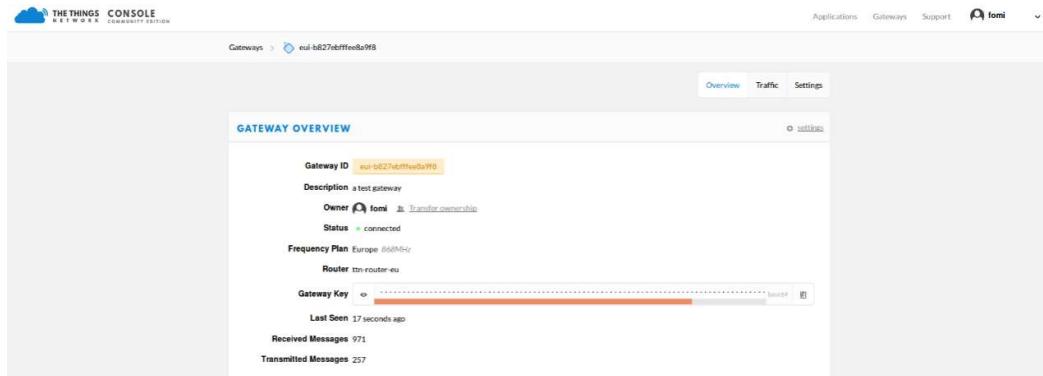
The screenshot shows the 'REGISTER GATEWAY' form in The Things Network Console. The 'Frequency Plan' section is expanded, displaying a list of frequency ranges for different regions:

- Europe: 868MHz
- Asia: 920-923MHz
- Asia: 923-925MHz
- Australia: 915MHz
- China: 470-510MHz
- Europe: 868MHz
- India: 865-867MHz
- Korea: 920-923MHz
- Russia: 864-870MHz
- United States: 915MHz

The screenshot shows the 'REGISTER GATEWAY' form in The Things Network Console. The 'Router' field has been populated with 'ttn-router-eu'.

**Note:** Remember to select the check box “I’m using the legacy packet forwarder”, and type the Gateway EUI which you have got in the step 1;

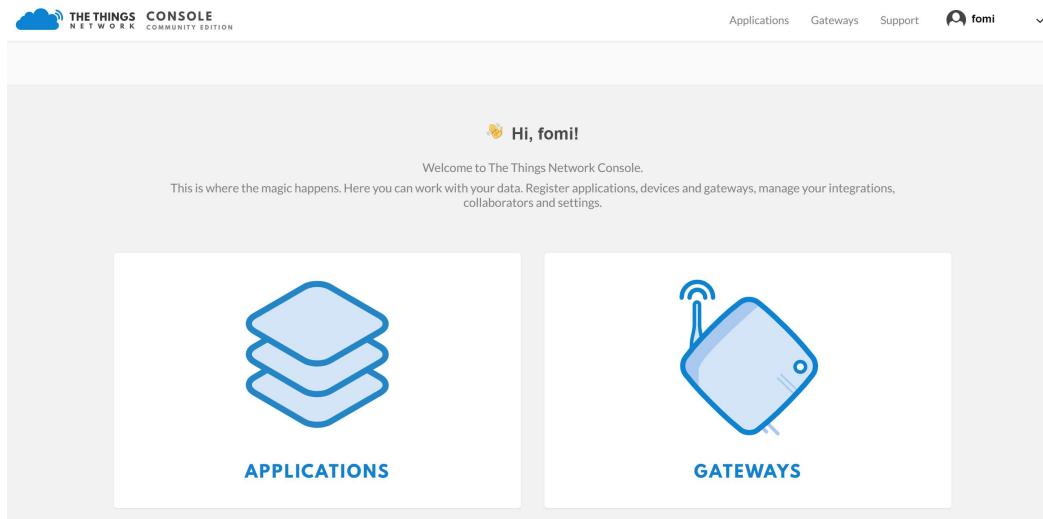
- 5) After the above operations, click “Register Gateway” in the bottom right corner of this page to register your gateway;
- 6) If register successfully, you will see the following page:



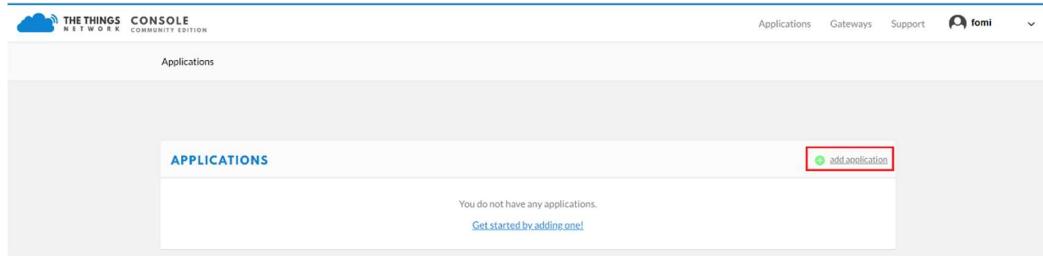
Please notice the “Status” item in the above page, “connected” indicates that your gateway has connected with TTN LoRa server successfully;

## Step 2: Add a LoRa node on TTN LoRa application server.

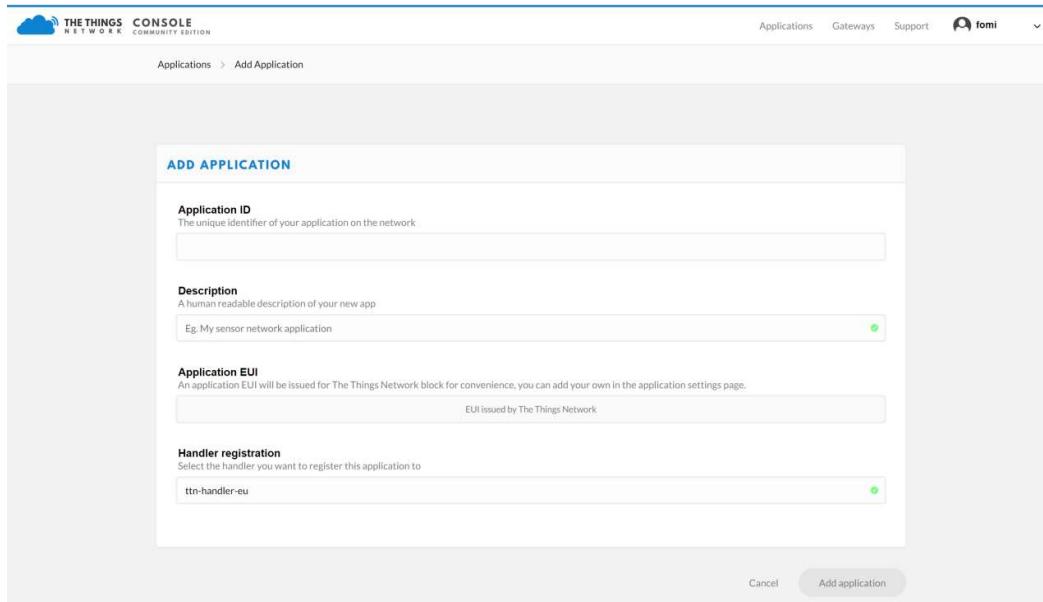
- 1) Login <https://www.thethingsnetwork.org/>, and open the Console panel:



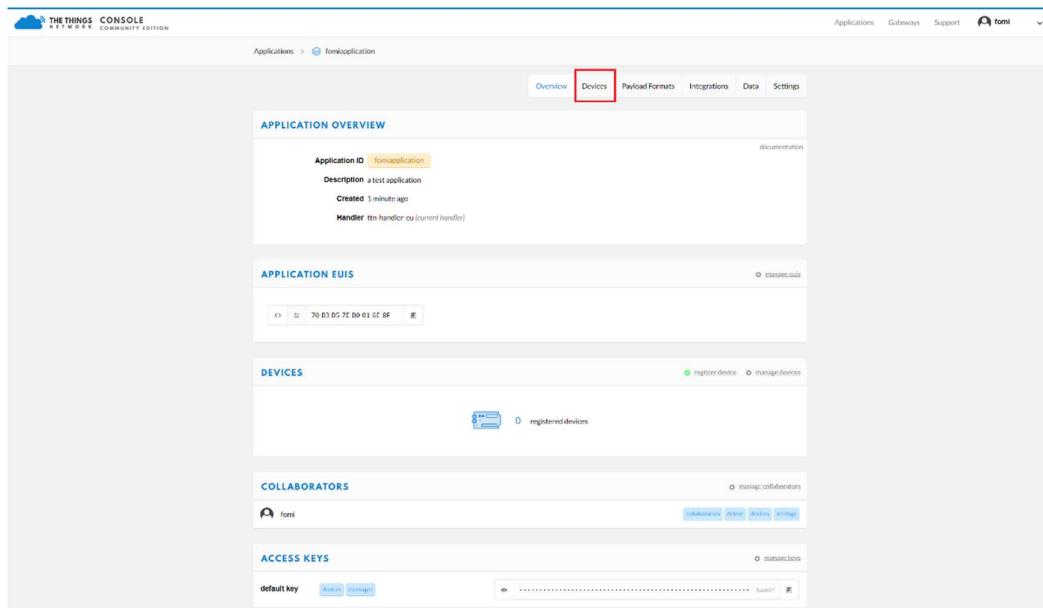
- 2) Click on “APPLICATIONS”, then you can see the following page. Click “add application”:



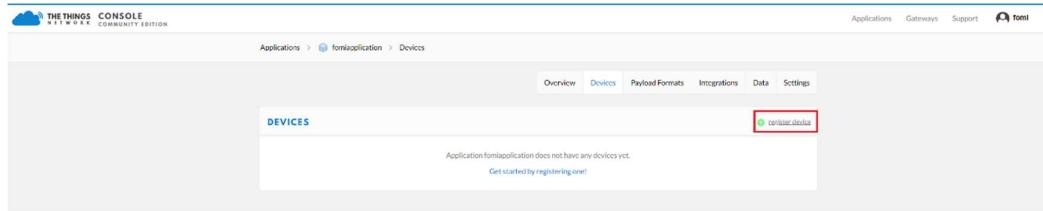
3) Fill in the following items, and click “Add application”:



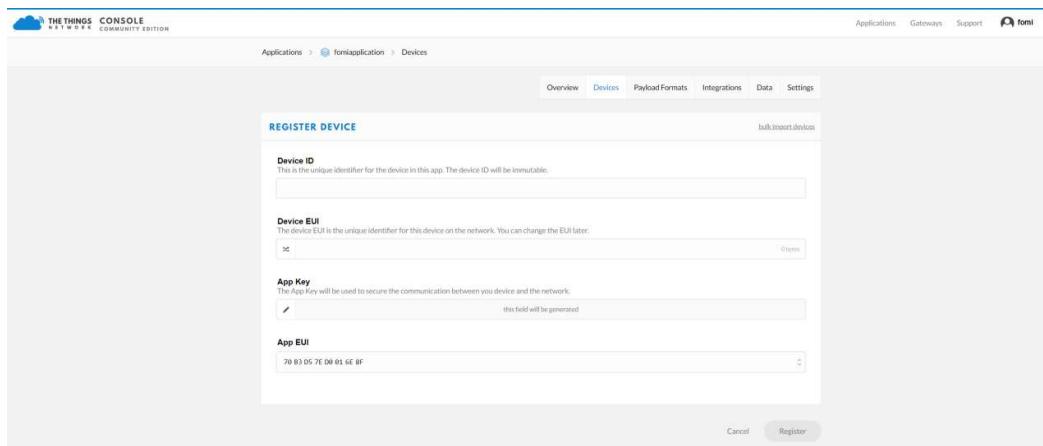
4) You'll see the following page, and click “Devices” as follow:



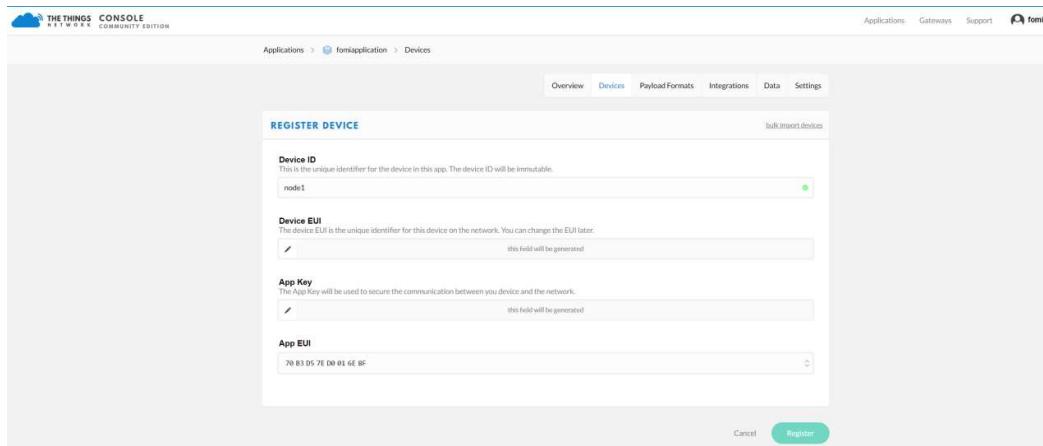
5) Click “register device”:



6) Fill in them:



Note: You can generate a Device EUI automatically by click



7) After clicking “Register”, you will see the following page. Please remember “Device EUI”, “Application EUI”, and “App Key”, because you will use them to configure your LoRa node in the step 3.

The screenshot shows the 'DEVICE OVERVIEW' section of the The Things Network Console. It displays the following information:

- Application ID:** fomiapplication
- Device ID:** node1
- Activation Method:** OTAA
- Device EU EUI:** 00 74 BD 5A 97 B3 68 90
- Application EU EUI:** 70 B3 D5 7E DB B1 6E 8F
- App Key:** (redacted)
- Status:** never seen
- Frames up:** 0 ([reset frame counters](#))
- Frames down:** 0

Below this, the 'DOWNLINK' section shows scheduling and payload settings. A green 'Send' button is at the bottom right.

### Step 3: Configure the LoRa node to join

- 1) Configure the LoRa node through AT command:

```
"at+set_config=dev_eui:xxxxxxxxxxxxxxxxxxxx"
"at+set_config=app_eui:xxxxxxxxxxxxxxxxxxxx"
"at+set_config=app_key:xxxxxxxxxxxxxxxxxxxx"
```

- 2) The "Status" indicates that the LoRa node has joined successfully:

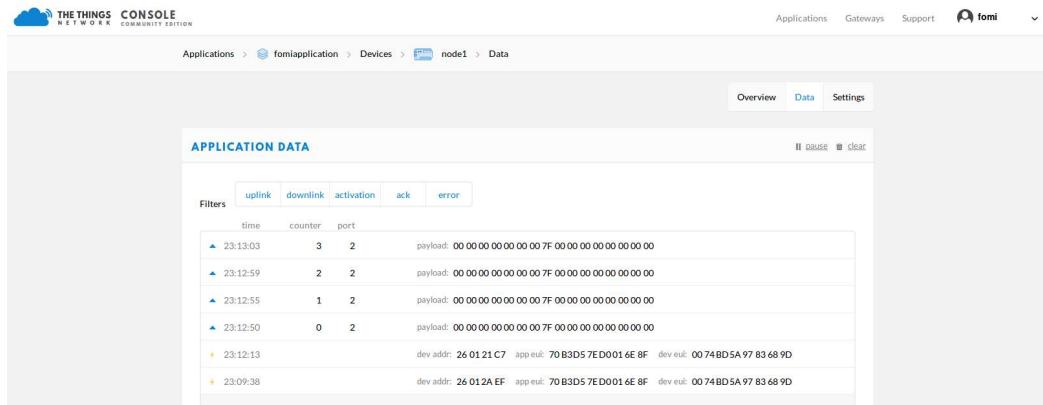
The screenshot shows the 'DEVICE OVERVIEW' section of the The Things Network Console. It displays the following information:

- Application ID:** fomiapplication
- Device ID:** node1
- Activation Method:** OTAA
- Device EU EUI:** 00 74 BD 5A 97 B3 68 90
- Application EU EUI:** 70 B3 D5 7E DB B1 6E 8F
- App Key:** (redacted)
- Device Address:** 20 01 2A EF
- Network Session Key:** (redacted)
- App Session Key:** (redacted)
- Status:** 53 seconds ago
- Frames up:** 0 ([reset frame counters](#))
- Frames down:** 0

- 3) Send some data from LoRa node to TTN:

```
at+send=0,2,00000000000000007F0000000000000000
```

You will see the data on TTN as follow:



The screenshot shows the 'Data' tab of the The Things Network Console. The URL in the address bar is 'Applications > fomiapplication > Devices > node1 > Data'. The main area is titled 'APPLICATION DATA' and contains a table of received messages. The columns are 'time', 'counter', 'port', 'payload', 'dev addr', 'app eui', and 'dev eui'. The table shows the following data:

time	counter	port	payload	dev addr	app eui	dev eui
23:13:03	3	2	00 00 00 00 00 00 7F 00 00 00 00 00 00 00 00			
23:12:59	2	2	00 00 00 00 00 00 7F 00 00 00 00 00 00 00 00			
23:12:55	1	2	00 00 00 00 00 00 7F 00 00 00 00 00 00 00 00			
23:12:50	0	2	00 00 00 00 00 00 7F 00 00 00 00 00 00 00 00			
23:12:13				26 01 21 C7	70 B3D5 7ED001 6E 8F	00 74 BD 5A 97 83 68 9D
23:09:38				26 01 2A EF	70 B3D5 7ED001 6E 8F	00 74 BD 5A 97 83 68 9D

## Postscript Note

This tutorial is simplest for LoRa beginners, and the default OS of the LoRa gateway is LoRa gateway OS which is based on Yocto.

If you want to use Raspbian OS on the LoRa gateway, please see the next tutorial, **From zero to build a LoRa gateway using RAK2243, Raspberry Pi, and Raspbian OS**. It is an advanced tutorial for the RAK LoRa Develop Kit.

## Special Announcement

The LoRa server and the LoRa application server of the RAK LoRa Develop Kit are both based on a great open source project which is made by Orne Brocaar.

You can find more about it in the loraserver official website:

<https://www.loraserver.io/>.

You can find the open source project on Github here:

<https://github.com/brocaar?tab=repositories>

