Connecting RAKwireless Commercial Gateways to the Cloud

AWS + ChirpStack

Version V1.0 | April 2020

www.RAKwireless.com

Visit our website for more document.

Table of Contents

1 Prerequisites

This document assumes you have a running Ubuntu on an AWS EC2 instance and you are able to access it via an SSH terminal.

In case you have not yet deployed one please refer to the document below:

Deployment and Management of Ubuntu on an AWS EC2 Instance

2 Installing ChirpStack

It is always a good idea to make an update and upgrade of your packages. In order to do so run the following commands in the terminal:

```
sudo apt update
```

```
sudo apt upgrade
```

After the procedure is completed, we are going to install ChirpStack and its dependencies. To do this first we need to install Git with the command:

```
sudo apt install git
```

Next, we clone the RAKwireless Ubuntu ChirpStack repository:

```
git clone https://github.com/RAKWireless/install_ChirpStack_on_ubuntu.git
```

After the cloning is complete open the newly created folder with the command:

```
cd install_ChirpStack_on_ubuntu
```

Run the installation script:

```
sudo ./install.sh
```

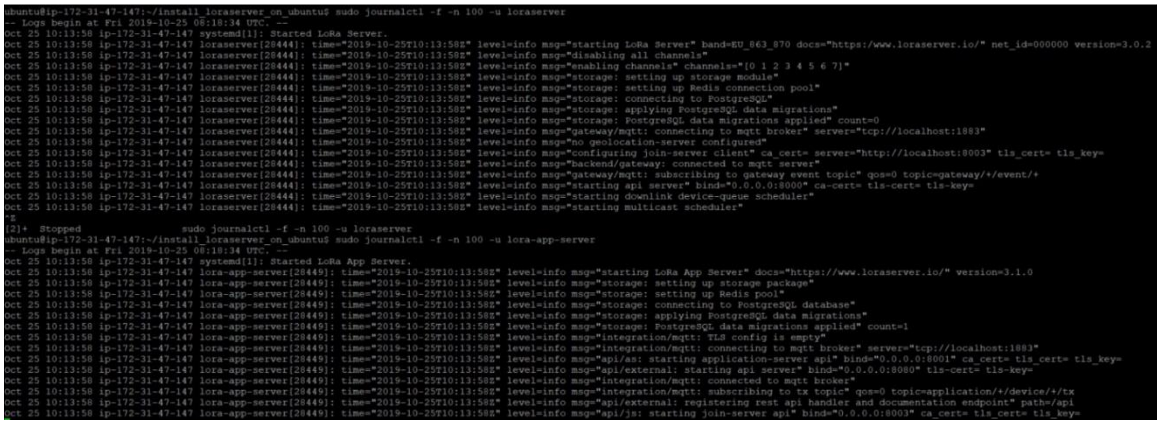
After the installation is completed check if all went well by executing the commands:

```
sudo journalctl -f -n 100 -u lora-app-server
```

```
sudo journalctl -f -n 100 -u ChirpStack
```

You should see no errors as in Figure 1. Make sure you interrupt the output of the commands above with the key combination “Ctrl+z” so you can continue with the configuration process.

In case you want to use the Semtech Packet Forwarder to connect your Gateway to the LoRa Network Server proceed to **Section 3**. If you are going to use the MQTT Bridge instead proceed directly to **Section 4**.



```
ubuntu@ip-172-31-47-147:~/install_loraserver_on_ubuntu$ sudo journalctl -f -n 100 -u loraserver
-- Logs begin at Fri 2019-10-25 08:18:34 UTC. --
Oct 25 10:13:58 ip-172-31-47-147 systemd[1]: Started LoRa Server.
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="starting LoRa Server" band=EU_863_870 docs="https:/www.loraserver.io/" net_id=000000 version=3.0.2
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="disabling all channels"
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="enabling channels" channels="[0 1 2 3 4 5 6 7]"
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="storage: setting up storage module"
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="storage: setting up Redis connection pool"
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="storage: connecting to PostgreSQL"
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="storage: applying PostgreSQL data migrations"
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="storage: PostgreSQL data migrations applied" count=0
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="gateway/mqtt: connecting to mqtt broker" server="tcp://localhost:1883"
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="no geolocation-server configured"
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="configuring join-server client" ca_cert= server="http://localhost:8003" tls_cert= tls_key=
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="backend/gateway: connected to mqtt server"
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="gateway/mqtt: subscribing to gateway event topic" qos=0 topic=gateway/+/event/+
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="starting api server" bind="0.0.0.0:8000" ca-cert= tls-cert= tls-key=
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="starting downlink device-queue scheduler"
Oct 25 10:13:58 ip-172-31-47-147 loraserver[28444]: time="2019-10-25T10:13:58Z" level=info msg="starting multicast scheduler"
^Z
[2]+  Stopped                 sudo journalctl -f -n 100 -u loraserver
ubuntu@ip-172-31-47-147:~/install_loraserver_on_ubuntu$ sudo journalctl -f -n 100 -u lora-app-server
-- Logs begin at Fri 2019-10-25 08:18:34 UTC. --
Oct 25 10:13:58 ip-172-31-47-147 systemd[1]: Started LoRa App Server.
Oct 25 10:13:58 ip-172-31-47-147 lora-app-server[28449]: time="2019-10-25T10:13:58Z" level=info msg="starting LoRa App Server" docs="https://www.loraserver.io/" version=3.1.0
Oct 25 10:13:58 ip-172-31-47-147 lora-app-server[28449]: time="2019-10-25T10:13:58Z" level=info msg="storage: setting up storage package"
Oct 25 10:13:58 ip-172-31-47-147 lora-app-server[28449]: time="2019-10-25T10:13:58Z" level=info msg="storage: setting up Redis pool"
Oct 25 10:13:58 ip-172-31-47-147 lora-app-server[28449]: time="2019-10-25T10:13:58Z" level=info msg="storage: connecting to PostgreSQL database"
Oct 25 10:13:58 ip-172-31-47-147 lora-app-server[28449]: time="2019-10-25T10:13:58Z" level=info msg="storage: applying PostgreSQL data migrations"
Oct 25 10:13:58 ip-172-31-47-147 lora-app-server[28449]: time="2019-10-25T10:13:58Z" level=info msg="storage: PostgreSQL data migrations applied" count=1
Oct 25 10:13:58 ip-172-31-47-147 lora-app-server[28449]: time="2019-10-25T10:13:58Z" level=info msg="integration/mqtt: TLS config is empty"
Oct 25 10:13:58 ip-172-31-47-147 lora-app-server[28449]: time="2019-10-25T10:13:58Z" level=info msg="integration/mqtt: connecting to mqtt broker" server="tcp://localhost:1883"
Oct 25 10:13:58 ip-172-31-47-147 lora-app-server[28449]: time="2019-10-25T10:13:58Z" level=info msg="api/as: starting application-server api" bind="0.0.0.0:8001" ca_cert= tls_cert= tls_key=
Oct 25 10:13:58 ip-172-31-47-147 lora-app-server[28449]: time="2019-10-25T10:13:58Z" level=info msg="api/external: starting api server" bind="0.0.0.0:8080" tls-cert= tls-key=
Oct 25 10:13:58 ip-172-31-47-147 lora-app-server[28449]: time="2019-10-25T10:13:58Z" level=info msg="integration/mqtt: connected to mqtt broker"
Oct 25 10:13:58 ip-172-31-47-147 lora-app-server[28449]: time="2019-10-25T10:13:58Z" level=info msg="integration/mqtt: subscribing to tx topic" qos=0 topic=application/+/device/+/tx
Oct 25 10:13:58 ip-172-31-47-147 lora-app-server[28449]: time="2019-10-25T10:13:58Z" level=info msg="api/external: registering rest api handler and documentation endpoint" path=/api
Oct 25 10:13:58 ip-172-31-47-147 lora-app-server[28449]: time="2019-10-25T10:13:58Z" level=info msg="api/js: starting join-server api" bind="0.0.0.0:8003" ca_cert= tls_cert= tls_key=
```

Figure 1 | ChirpStack Journal Control Output (no errors)

3 Installing the Gateway Bridge

We are going to provide an outline on how to perform the installation. For detailed information you can visit ChirpStack.io.

Run the following commands command:

sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 1CE2AFD36DBCCA00

sudo echo "deb https://artifacts.ChirpStack.io/packages/3.x/deb stable main" | *sudo tee /etc/apt/sources.list.d/ChirpStack.list*

sudo apt-get update

This will update the Ubuntu Repositories.

Proceed with installing the Bridge itself:

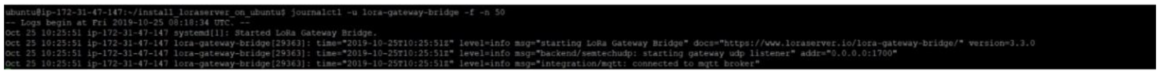
sudo apt-get install lora-gateway-bridge

Start the Bridge service:

sudo systemctl start lora-gateway-bridge

Check that it is working as it should:

journalctl -u lora-gateway-bridge -f -n 50



```
ubuntu@ip-172-31-47-147:~/install_loraserver_on_ubuntu$ journalctl -u lora-gateway-bridge -f -n 50
-- Logs begin at Fri 2019-10-25 08:18:34 UTC. --
Oct 25 10:25:51 ip-172-31-47-147 systemd[1]: Started LoRa Gateway Bridge.
Oct 25 10:25:51 ip-172-31-47-147 lora-gateway-bridge[29363]: time="2019-10-25T10:25:51Z" level=info msg="starting LoRa Gateway Bridge" docs="https://www.loraserver.io/lora-gateway-bridge/" version=3.3.0
Oct 25 10:25:51 ip-172-31-47-147 lora-gateway-bridge[29363]: time="2019-10-25T10:25:51Z" level=info msg="backend/semtechudp: starting gateway udp listener" addr="0.0.0.0:1700"
Oct 25 10:25:51 ip-172-31-47-147 lora-gateway-bridge[29363]: time="2019-10-25T10:25:51Z" level=info msg="integration/mqtt: connected to mqtt broker"
```

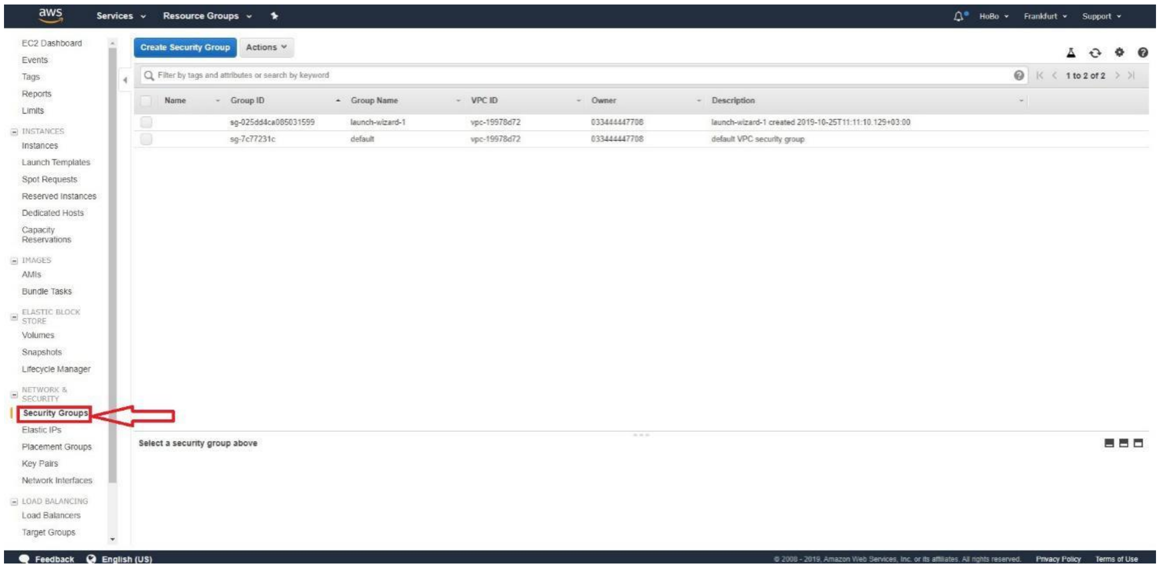
Figure 2 | Gateway Bridge Journal Control Output (no errors)

4 Configure the AWS Security

By default, all inbound traffic to an AWS Instance is blocked, only port 22 (SSH) is open. You need to add a set of rules in order for the Gateway and LoRa Network Server to be able to communicate:

- The Semtech Packet Forwarder needs **UDP port 1700;**
- MQTT Bridge (unsecured) needs **TCP port 1883**;
- MQTT Bridge (secured) needs **TCP port 8883**;
- ChirpStack Web Ui needs **TCP port 8080**.

Thus, open the Security Groups tab in the AWS Dashboard:

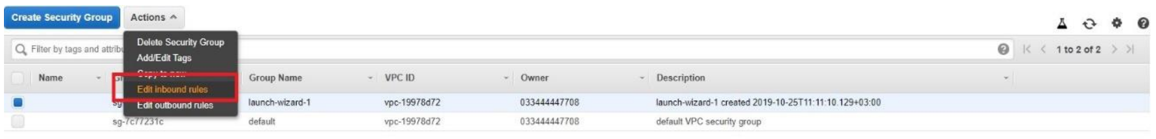


Name	Group ID	Group Name	VPC ID	Owner	Description
	sg-025dd4ca085031599	launch-wizard-1	vpc-19978d72	033444447708	launch-wizard-1 created 2019-10-25T11:11:10.129+03:00
	sg-7c77231c	default	vpc-19978d72	033444447708	default VPC security group

Figure 3 | AWS Security Groups

Select you desired Security Group (Ubuntu Instance). If you have multiple instances you can use the date and time of creation of the group as a guide to which is the one you want.

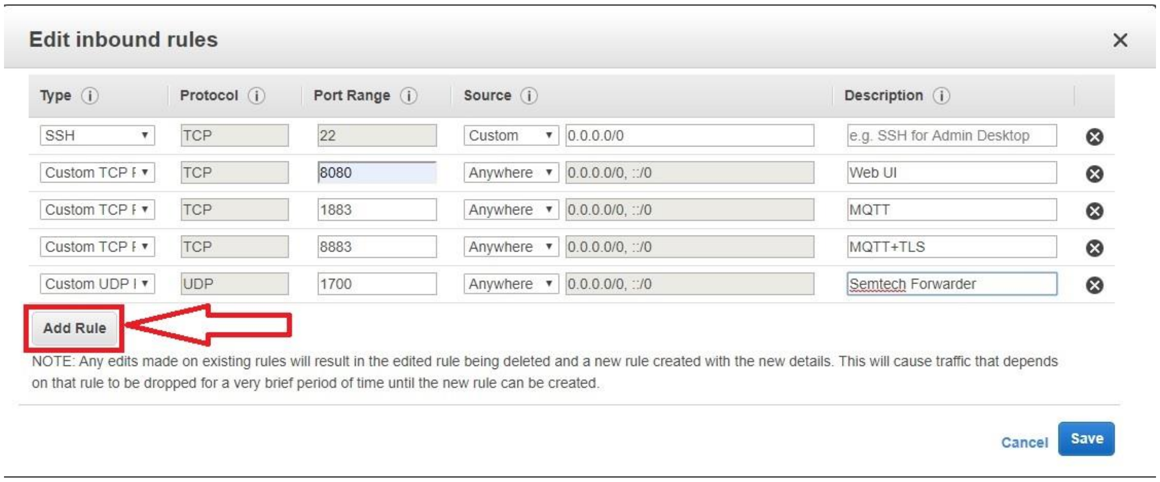
Click the "Action" button and from the drop-worn menu select **Edit Inbound Rules**:



Name	Group ID	Group Name	VPC ID	Owner	Description
	sg	launch-wizard-1	vpc-19978d72	033444447708	launch-wizard-1 created 2019-10-25T11:11:10.129+03:00
	sg-7c77231c	default	vpc-19978d72	033444447708	default VPC security group

Figure 16 | Security Group Inbound Rules

In the windows that has opened press the “Add Rule” button and add all the 4 rules we mentioned before (refer to Figure 4):


Edit inbound rules

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8080	Anywhere	0.0.0.0/0, ::/0	Web UI
Custom TCP F	TCP	1883	Anywhere	0.0.0.0/0, ::/0	MQTT
Custom TCP F	TCP	8883	Anywhere	0.0.0.0/0, ::/0	MQTT+TLS
Custom UDP I	UDP	1700	Anywhere	0.0.0.0/0, ::/0	Semtech Forwarder

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Figure 4 | Adding Inbound Rules

Note: It is good practice to name them in accordance with what each of them represents as in Figure 4.

Make sure to Save with the button in the lower right corner.

Finally check if the rules you created are working, by entering your instance Public IP address using port 8080 in a browser window. You should see the Login page of the ChirpStack Web UI (for example 3.120.237.38:8080 as in Figure 5).

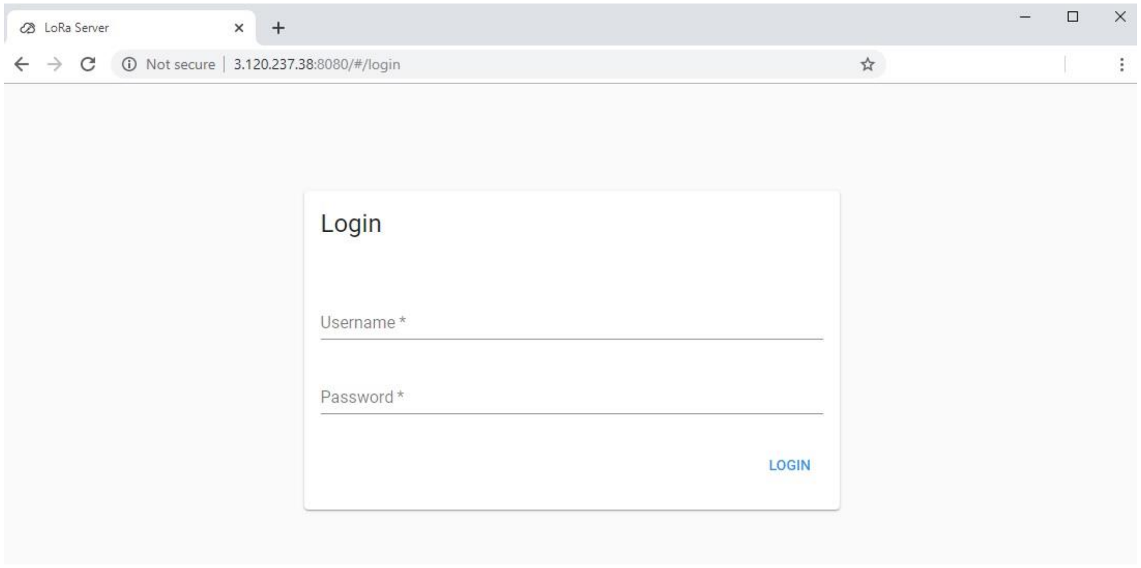


Figure 5| ChirpStack Login Page

Note: The default *Username/Password* are **admin/admin**. Additionally there are profiles crated in the RAKwireless ChirpStack installation, so you do not need to make those yourself and you can directly proceed to adding your Gateway.

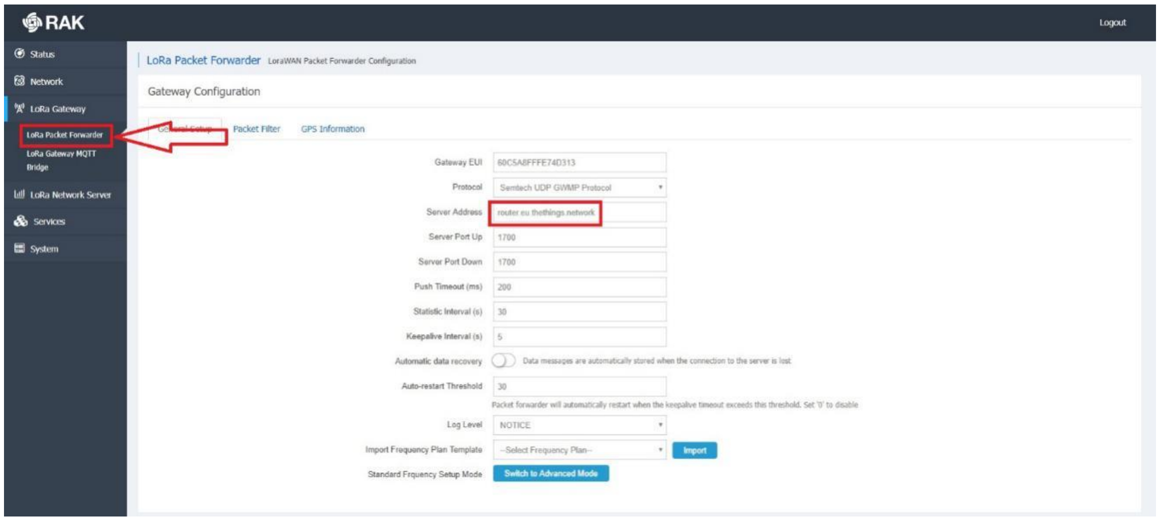
5 Configuring your Gateway

5.1 Configuring the Semtech Packet Forwarder

In the Gateway Web UI go to the LoRa Gateway tab and open the LoRa Packet Forwarder sub menu.

On details about the configuration options of the Gateway check the link.

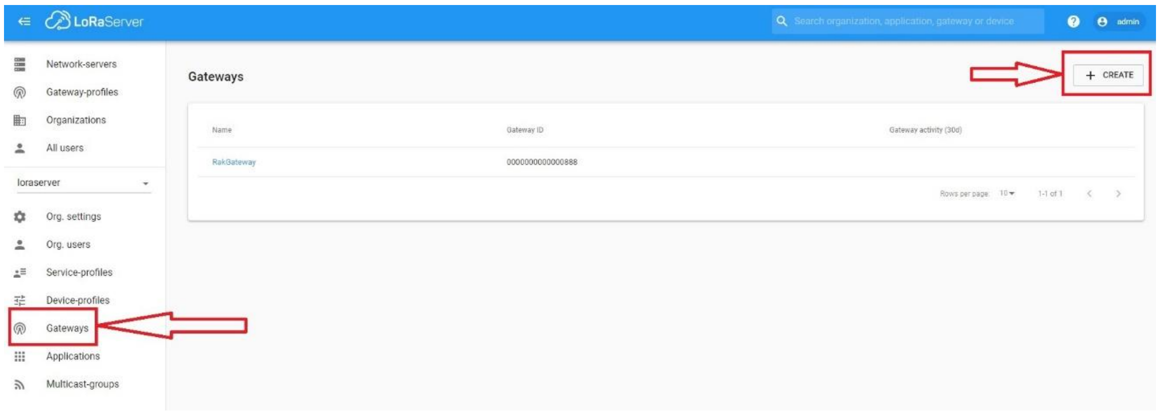
By default, the Gateway is set to point to TTN, using the Semtech Packet Forwarder. You need only change the Server Address in order to forward the traffic to your ChirpStack running on the Ubuntu Instance (AWS). Enter your Instance Public IP Address in the field marked with the red rectangle in Figure 6.



Gateway EUI: 60C5A8FFFE74D313
Protocol: Semtech UDP GWMP Protocol
Server Address: router.eu.thethings.network
Server Port Up: 1700
Server Port Down: 1700
Push Timeout (ms): 200
Statistic Interval (s): 30
Keepalive Interval (s): 5
Automatic data recovery: Data messages are automatically stored when the connection to the server is lost
Auto-restart Threshold: 30
Packet forwarder will automatically restart when the keepalive timeout exceeds this threshold. Set '0' to disable
Log Level: NOTICE
Import Frequency Plan Template: --Select Frequency Plan--
Standard Frquency Setup Mode

Figure 6 | ChirpStack Packet Forwarder Configuration

Save and Apply the changes and go to your ChirpStack Web UI running on the AWS Instance (IP Address:8080). Go to the Gateway tab. Press the "Create" button.

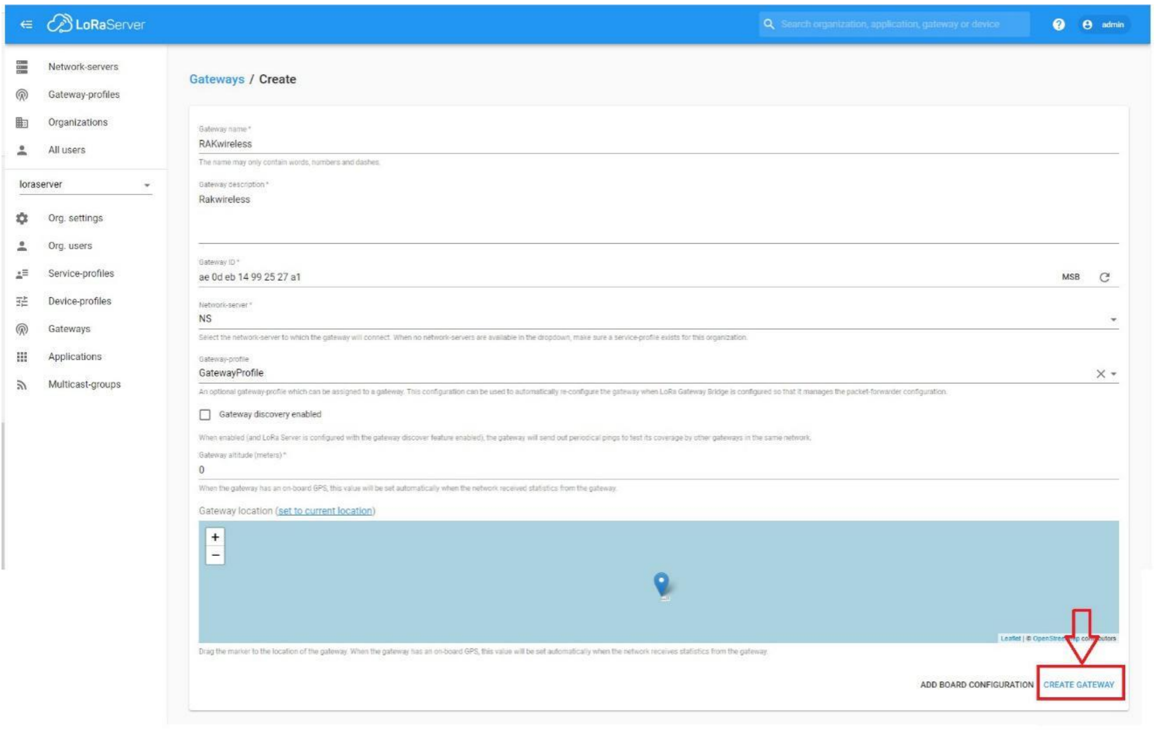


Gateways

Name	Gateway ID	Gateway activity (30d)
RakGateway	0000000000000888	

Figure 7 | ChirpStack Gateways Creation

In the next window input the Gateway Name, EUI and Description. Select a network server and Service Profile from the drop-down menu (remember those are pre-configured with the RAKwireless image). Finally click the “Create Gateway” button.



Gateways / Create

Gateway name *
RAKwireless
The name may only contain words, numbers and dashes.

Gateway description *
Rakwireless

Gateway ID *
ae 0d eb 14 99 25 27 a1

Network-server *
NS
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
GatewayProfile
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.

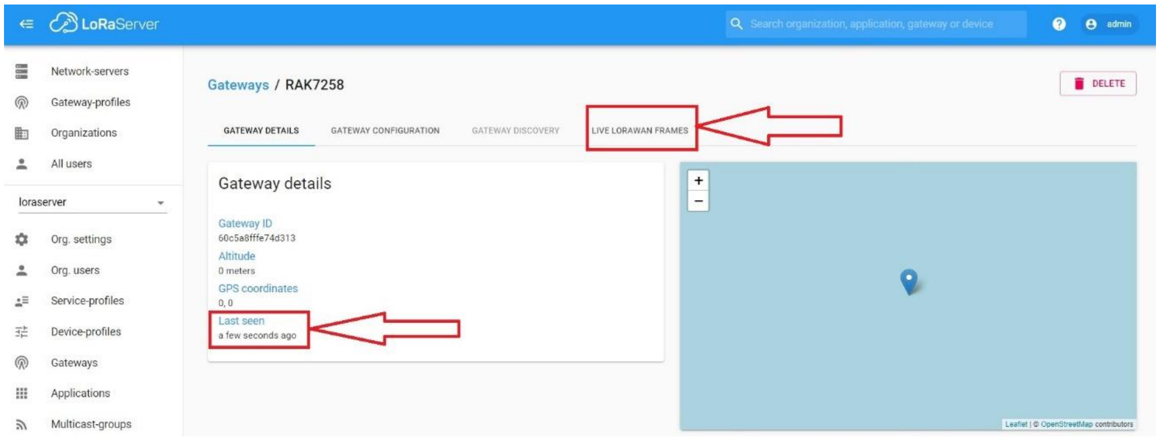
Gateway altitude (meters) *
0
When the gateway has an on-board GPS, this value will be set automatically when the network received statistics from the gateway.

Gateway location (set to current location)

Drag the marker to the location of the gateway. When the gateway has an on-board GPS, this value will be set automatically when the network receives statistics from the gateway.

Figure 8 | ChirpStack Gateway Parameters

Assuming you entered the parameters correctly you should see your Gateway status as seen is a few second in the Gateway Details tab (Figure 9). You can also monitor Live LoRa Frames in the tab with the same name to see incoming traffic.



Gateway details

Gateway ID
60c5a8fffe74d313

Altitude
0 meters

GPS coordinates
0, 0

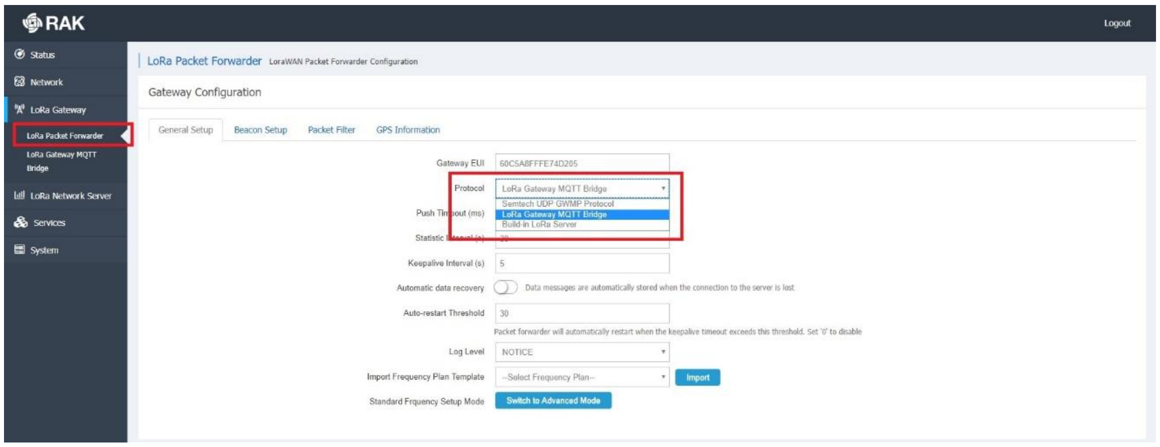
Last seen
a few seconds ago

Figure 9 | ChirpStack Gateway Details

5.2 Configuring the MQTT Bridge

If you want to use the MQTT Bridge to forward your LoRa Traffic to your LoRa Network Server you need to configure your Gateway use the Bridge instead of the Packet Forwarder.

Go back to the Gateway Web UI and go to the Packet Forwarder sub-menu. Change the protocol from the drop-down menu to LoRa Gateway MQTT Bridge. Save and Apply



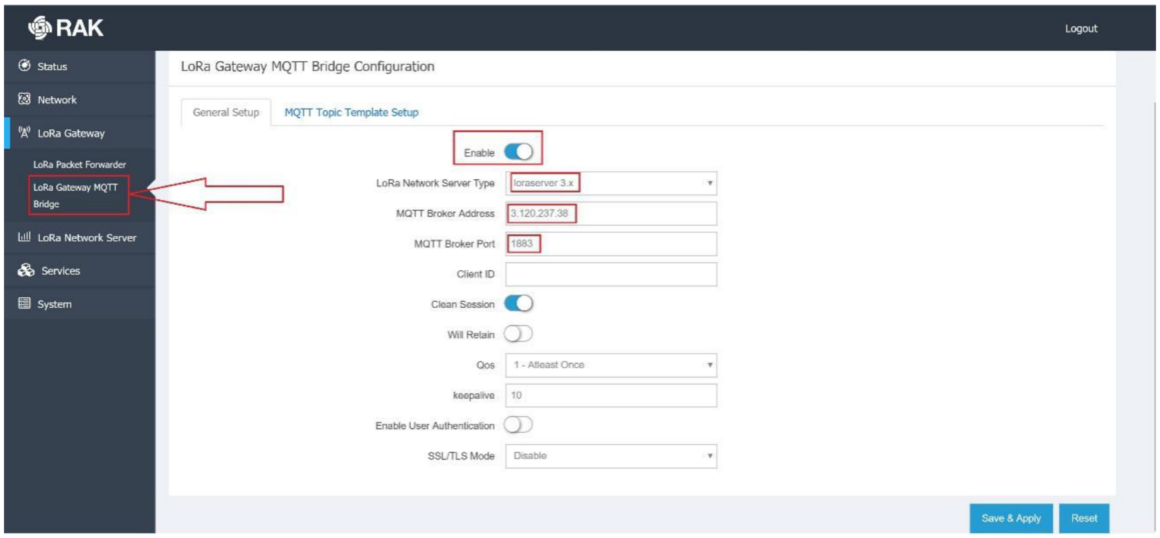
LoRa Packet Forwarder LoraWAN Packet Forwarder Configuration

Gateway Configuration

Gateway EUI: 60C5A8FFFE74D205
Protocol: LoRa Gateway MQTT Bridge
Push Timeout (ms)
Statistic Interval (s): 30
Keepalive Interval (s): 5
Automatic data recovery: Data messages are automatically stored when the connection to the server is lost
Auto-restart Threshold: 30
Packet forwarder will automatically restart when the keepalive timeout exceeds this threshold. Set '0' to disable
Log Level: NOTICE
Import Frequency Plan Template: --Select Frequency Plan--
Standard Frquency Setup Mode

Figure 10 | Gateway MQTT Bridge Protocol

Next go to the LoRa Gateway MQTT Bridge sub-menu. Enable the functionality by with the blue slider and choose the type of LoRa Network Server you are going to be using (this is important as there is difference in the MQTT topic templates, JSON vs Protobuf, for 2x vs 3x respectively). Lastly set the address to the address of the AWS Instance and the port to 1883, Save and Apply.



LoRa Gateway MQTT Bridge Configuration

Enable	On
LoRa Network Server Type	loraserver 3.x
MQTT Broker Address	3.120.237.38
MQTT Broker Port	1883
Client ID	
Clean Session	On
Will Retain	Off
Qos	1 - Atleast Once
keepalive	10
Enable User Authentication	Off
SSL/TLS Mode	Disable

Figure 11 | Gateway MQTT Bridge Parameters

Lastly you need to add the Gateway in ChirpStack if you haven't already done so (for example if you used it with the Packet Forwarder before). This is done the same way as it was in **Section 5**.

This concludes the tutorial.