目录

《使用前说明》

本示例仅作为演示如何使用我司 LoRaWAN 节点模块和 LoRaWAN 网关接入服务器建立 LoRaWAN 网络。

说明：

1、LoRaWAN 服务器环境需要客户自己搭建，对服务器环境搭建（包括本示例中的 chirpstack 服务器的环境搭建）我司暂不提供技术支持，可自行参考官方文档。

2、本示例中使用 chirpstack v4 开源服务器进行演示，由于 chirpstack 服务器版本会存在升级或其他原因，本示例中涉及 chirpstack 服务器搭建内容有可能无法适配后续更新后的 chirpstack 服务器的搭建，可自行参考官方文档。

3、在示例 Ubuntu 中搭建 chirpstack 服务器可能会出现类似“下列软件包有未满足的依赖关系”等问题，需要客户自己解决处理，关于服务器搭建中产生的问题我司暂不提供技术支持，可自行参考官方文档。

4、本示例使用节点模块型号为：E78-400TBL-02（搭载了 E78-470LN22S(6601)模组的测试板）；网关型号为：E870-L470LG11。

chirpstack 官方网址：

https://www.chirpstack.io/

E78-400TBL-02 购买地址：

https://detail.tmall.com/item.htm?id=609798136700&skuId=4860568252176

E870-L470LG11 购买地址：

https://detail.tmall.com/item.htm?id=667696115203&skuId=4850172833243

一、 Chirpstack 服务器简介和搭建

注：本文是参考 ChirpStack 官网 Ubuntu 系统安装方式，官网链接：

Chirpstack 是一款多组件的、部署简单的开源服务器，同时也是使用最广泛的 LoRaWAN 服务器。本次安装使用 Ubuntu 22.04。需要使用到的软件有 vim，请自行安装。

终端中输入 vim --version 命令来检查 Vim 版本，如安装，则返回版本号。

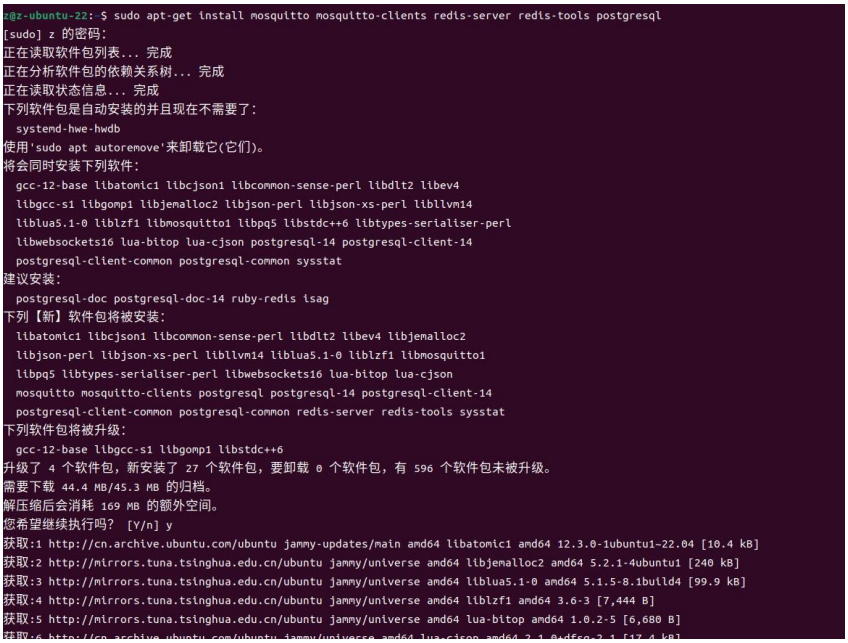
1、安装依赖项

MQTT 代理：一种发布/订阅协议，允许用户在其他人可以订阅的主题下发布信息。Mosquitto 是 MQTT 协议的流行实现。

Redis：一个内存数据库，用于存储相对短暂的数据。

PostgreSQL：开源软件包使用的长期存储数据库。

在 Ubuntu 终端输入以下指令安装：sudo apt-get install mosquitto mosquitto-clients redis-server redis-tools postgresql



等待安装完成后可以输入以下指令查询是否安装成功，如果安装成功返回版本号：

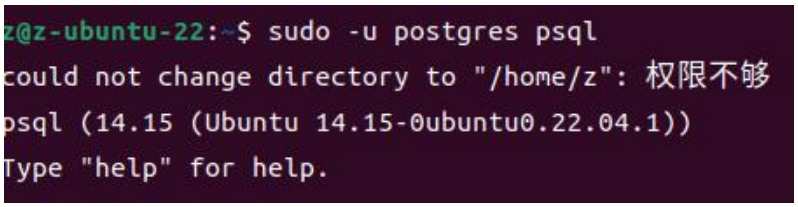
（1）验证 Mosquitto 安装成功：mosquitto -v

（2）验证 Redis 服务器安装成功：redis-cli --version

（3）PostgreSQL 数据库：到这里该数据库还未安装完成，本环节不做验证

2、设置 PostgreSQL 数据库和用户

（1）进入 PostgreSQL 的命令行实用程序：sudo -u postgres psql



在此提示符中，执行以下查询以设置 ChirpStack 堆栈组件使用的数据库。建议更改用户名和密码。请记住在更新和配置文件时使用这些其他值。由于这两个应用程序都使用同一个表来跟踪数据库升级，因此它们必须具有单独的数据库。

（2）通过以下指令创建了 role 为 chirpstack 密码都为 chirpstack：

create role chirpstack with login password 'chirpstack';

```
postgres=# create role chirpstack with login password 'chirpstack';
CREATE ROLE
```

（3）通过以下指令为服务器创建数据库：

create database chirpstack with owner chirpstack;

```
postgres=# create database chirpstack with owner chirpstack;
CREATE DATABASE
```

注意：从左到右，此处的第一个 chirpstack 代表数据库名称，第二个代表账号。

（4）\c chirpstack

```
postgres=# \c chirpstack
You are now connected to database "chirpstack" as user "postgres".
```

（5）启用 pg_trgm

create extension pg_trgm;

```
loraserver_as=# create extension pg_trgm;
CREATE EXTENSION
```

（6）离开数据库：\q

```
chirpstack=# \q
z@z-ubuntu-22:~$
```

3、设置 ChirpStack 软件存储库

（1）ChirpStack 提供了与 Ubuntu apt 包系统兼容的存储库。首先确保已安装：dirmngr 和 apt-transport-https，安装指令：sudo apt install apt-transport-https dirmngr

```
z@z-ubuntu-22:~$ sudo apt install apt-transport-https dirmngr
正在读取软件包列表... 完成
正在分析软件包的依赖关系树... 完成
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了：
  systemd-hwe-hwdb
使用'sudo apt autoremove'来卸载它(它们)。
将会同时安装下列软件：
  gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client gpg-wks-server
  gpgconf gpgsm gpgv
建议安装：
  tor parcimonie xloadimage scdaemon
下列【新】软件包将被安装：
  apt-transport-https
下列软件包将被升级：
  dirmngr gnupg gnupg-l10n gnupg-utils gpg gpg-agent gpg-wks-client
  gpg-wks-server gpgconf gpgsm gpgv
升级了 11 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 583 个软件包未被
升级。
需要下载 1,510 B/2,248 kB 的归档。
解压缩后会消耗 170 kB 的额外空间。
```

（2）设置此新存储库的密钥：sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 1CE2AFD36DBCCA00

```
z@z-ubuntu-22:~$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 1
CE2AFD36DBCCA00
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (s
ee apt-key(8)).
Executing: /tmp/apt-key-gpghome.yG00r5C6Z7/gpg.1.sh --keyserver keyserver.ubuntu
.com --recv-keys 1CE2AFD36DBCCA00
gpg: 密钥 1CE2AFD36DBCCA00：公钥 “Orne Brocaar <info@brocaar.com>” 已导入
gpg: 处理的总数：1
gpg:               已导入：1
z@z-ubuntu-22:~$
```

（3）通过创建新文件将存储库添加到存储库列表：

sudo echo "deb https://artifacts.chirpstack.io/packages/4.x/deb stable main" | sudo tee /etc/apt/sources.list.d/chirpstack.list

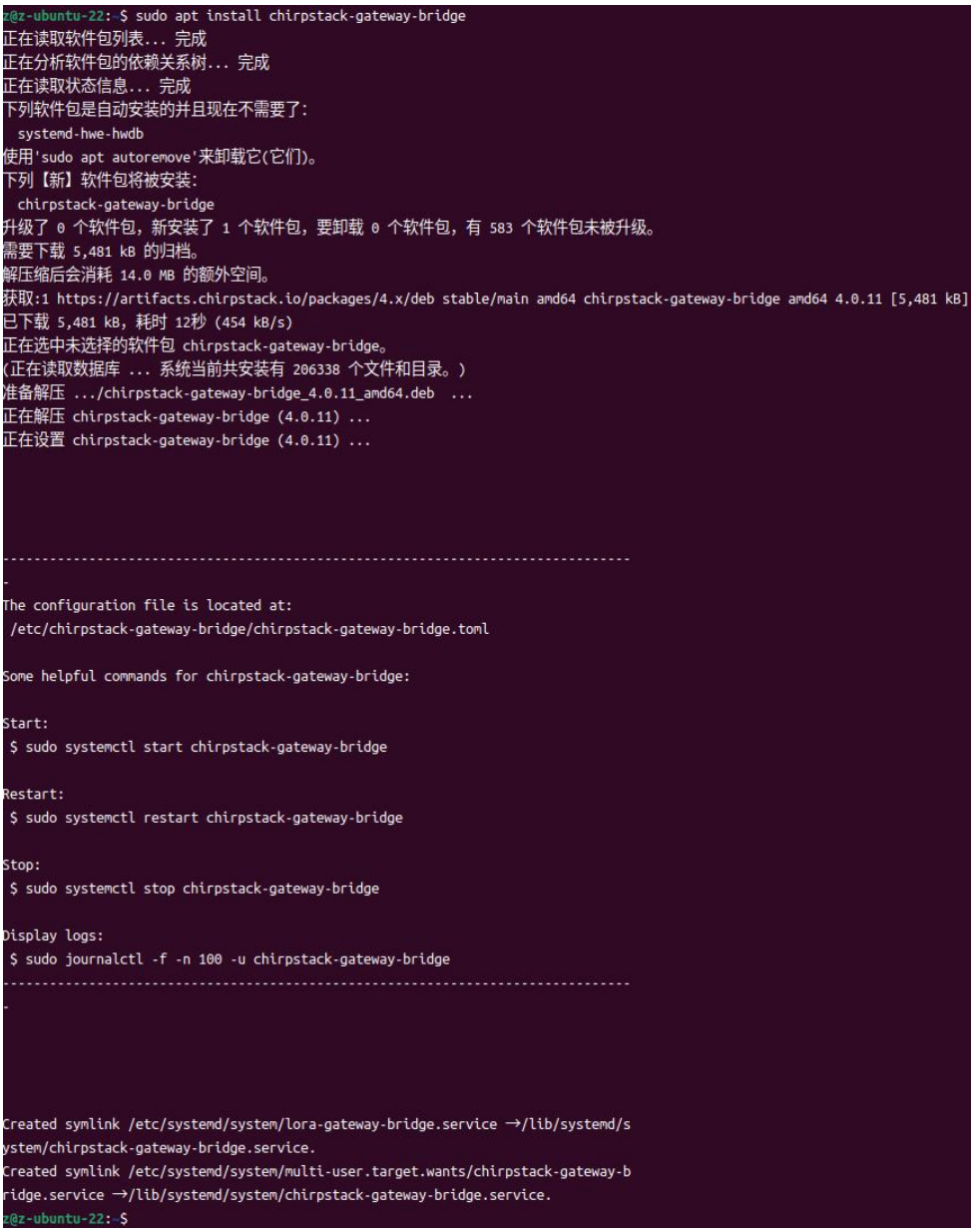
```
z@z-ubuntu-22:~$ sudo echo "deb https://artifacts.chirpstack.io/packages/4.x/deb
 stable main" | sudo tee /etc/apt/sources.list.d/chirpstack.list
deb https://artifacts.chirpstack.io/packages/4.x/deb stable main
z@z-ubuntu-22:~$
```

（4）更新 apt 包缓存：sudo apt update

4、安装 ChirpStack 网关网桥

注意：如果您打算仅在网关本身上运行 ChirpStack 网关桥，则可以跳过此步骤。

（1）使用以下命令安装网关网桥：sudo apt install chirpstack-gateway-bridge



注意：配置文件位于/etc/chirpstack-gateway-bridge/chirpstack-gateway-bridge.toml，请更新该部分以匹配适用于此 ChirpStack Gateway Bridge 实例的区域。

打开命令：sudo vim /etc/chirpstack-gateway-bridge/chirpstack-gateway-bridge.toml

（2）启动 ChirpStack 网关网桥服务：

sudo systemctl start chirpstack-gateway-bridge

sudo systemctl enable chirpstack-gateway-bridge

5、安装 ChirpStack 网络服务器

（1）使用以下指令安装软件包：sudo apt install chirpstack

```
z@z-ubuntu-22:~$ sudo apt install chirpstack
正在读取软件包列表... 完成
正在分析软件包的依赖关系树... 完成
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了:
  libflashrom1 libftdi1-2 libllvm13
使用'sudo apt autoremove'来卸载它(它们)。
建议安装:
  redis
下列【新】软件包将被安装:
  chirpstack
升级了 0 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 3 个软件包未被升级。
需要下载 11.6 MB 的归档。
解压缩后会消耗 50.1 MB 的额外空间。
获取:1 https://artifacts.chirpstack.io/packages/4.x/deb stable/main amd64 chirpstack amd64 4.10.2 [11.6 MB]
已下载 11.6 MB，耗时 14秒 (833 kB/s)
正在选中未选择的软件包 chirpstack。
(正在读取数据库 ... 系统当前共安装有 207744 个文件和目录。)
准备解压 .../chirpstack_4.10.2_amd64.deb  ...
正在解压 chirpstack (4.10.2) ...
正在设置 chirpstack (4.10.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/chirpstack.service →
 /lib/systemd/system/chirpstack.service.
z@z-ubuntu-22:~$
```

注意：该配置文件位于：/etc/chirpstack/chirpstack.toml，其中包含全局配置文件和各种区域配置文件，进入指令为：sudo vim /etc/chirpstack/chirpstack.toml

（2）启动 ChirpStack 网络服务器服务：

sudo systemctl start chirpstack

sudo systemctl enable chirpstack

（3）打印 ChirpStack 网络服务器日志输出：sudo journalctl -f -n 10 -u chirpstack

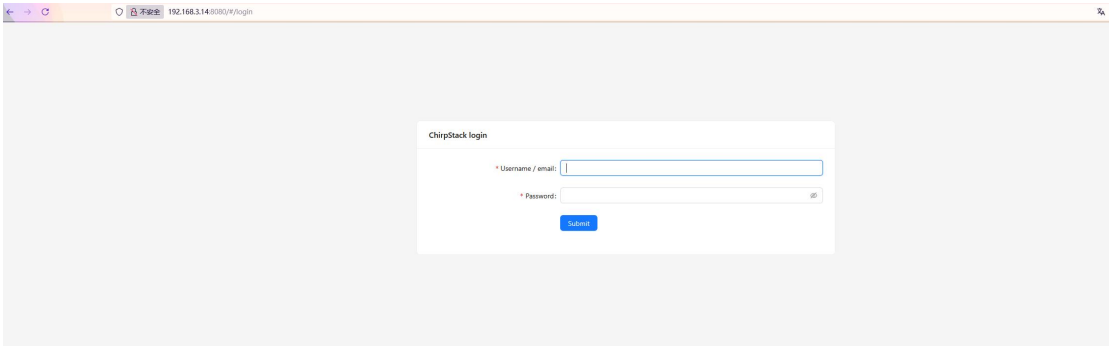
```
z@z-ubuntu-22:~$ sudo systemctl start chirpstack
z@z-ubuntu-22:~$ sudo systemctl enable chirpstack
z@z-ubuntu-22:~$ sudo journalctl -f -n 10 -u chirpstack
1月 05 17:03:54 z-ubuntu-22 chirpstack[2925]: 2025-01-05T09:03:54.329976Z  INFO chirpstack::gateway::backend:
Setting up gateway backend for region region_id=cn779 region_common_name=CN779
1月 05 17:03:54 z-ubuntu-22 chirpstack[2925]: 2025-01-05T09:03:54.329999Z  INFO chirpstack::gateway::backend::
mqtt: Starting MQTT event loop
1月 05 17:03:54 z-ubuntu-22 chirpstack[2925]: 2025-01-05T09:03:54.330341Z  INFO chirpstack::gateway::backend::
mqtt: Subscribing to gateway event topic region_id=au915_0 event_topic=$share/chirpstack/au915_0/gateway/+/eve
nt/+
1月 05 17:03:54 z-ubuntu-22 chirpstack[2925]: 2025-01-05T09:03:54.339466Z  INFO chirpstack::gateway::backend::
mqtt: Connecting to MQTT broker region_id=cn779 server_uri=tcp://localhost:1883 clean_session=false client_id=
b5d330d2b9983919
1月 05 17:03:54 z-ubuntu-22 chirpstack[2925]: 2025-01-05T09:03:54.339566Z  INFO chirpstack::downlink: Setting
up Class-B/C scheduler loop
1月 05 17:03:54 z-ubuntu-22 chirpstack[2925]: 2025-01-05T09:03:54.339572Z  INFO chirpstack::downlink: Setting
up multicast scheduler loop
1月 05 17:03:54 z-ubuntu-22 chirpstack[2925]: 2025-01-05T09:03:54.339586Z  INFO chirpstack::gateway::backend::
mqtt: Starting MQTT event loop
1月 05 17:03:54 z-ubuntu-22 chirpstack[2925]: 2025-01-05T09:03:54.339588Z  INFO chirpstack::api: Setting up AP
I interface bind=0.0.0.0:8080
1月 05 17:03:54 z-ubuntu-22 chirpstack[2925]: 2025-01-05T09:03:54.340243Z  INFO chirpstack::gateway::backend::
mqtt: Subscribing to gateway event topic region_id=cn779 event_topic=$share/chirpstack/cn779/gateway/+/event/+
1月 05 17:03:54 z-ubuntu-22 chirpstack[2925]: 2025-01-05T09:03:54.353613Z  INFO chirpstack::api::backend: Back
end interfaces API interface is disabled
```

6、完成安装，验证安装结果

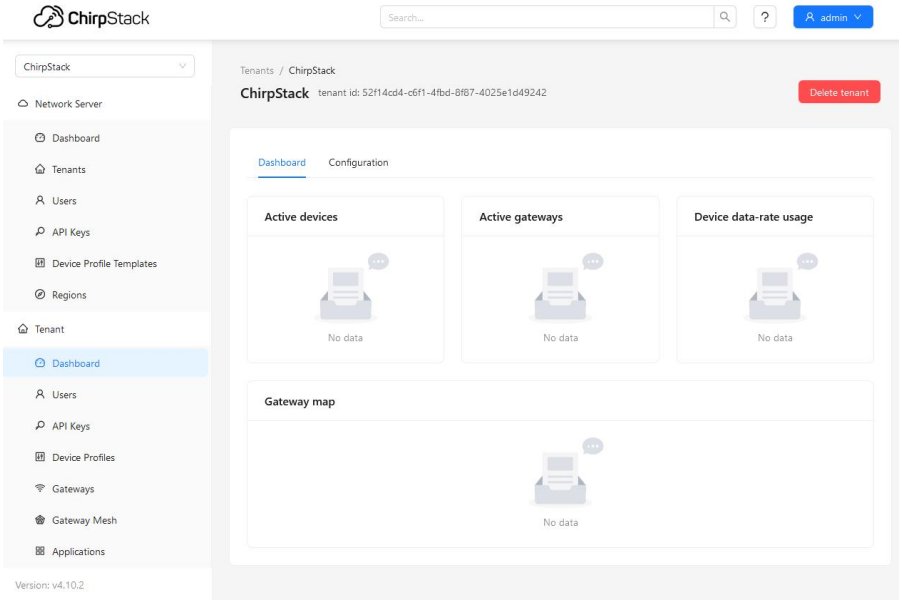
到此 ChirpStack 服务器基本搭建完成。我们可通过 IP 地址加上“:8080”来导航到 ChirpStack application server Web 界面。

IP 地址可以通过 ip a 查询：

```
z@z-ubuntu-22:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:ef:de:1f brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.3.14/24 brd 192.168.3.255 scope global dynamic noprefixroute ens33
       valid_lft 86355sec preferred_lft 86355sec
    inet6 240e:39b:f7a:5d72:a446:b48e:ca55:21/128 scope global dynamic noprefixroute
       valid_lft 7157sec preferred_lft 3557sec
    inet6 240e:39b:f7a:5d72:153b:c6aa:1eb3:60f3/64 scope global temporary dynamic
       valid_lft 7158sec preferred_lft 3558sec
    inet6 240e:39b:f7a:5d72:7659:43c7:697e:3eb4/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 7158sec preferred_lft 3558sec
    inet6 fe80::38df:d5f7:7a06:b056/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```



默认的账号密码均为 admin，登录后界面如下：



二、 使用 E78 节点+E870 网关接入 chirpstack

1、切换服务器地区频段

本次使用的演示样机为 470 频段的 E870-L470LG11 网关以及 E78-400TBL-02（搭载了 E78-470LN22S(6601)）节点模组，因此我们需要切换地区频段。

修改使用频段一共有两个文件中的参数需要进行修改，下述操作以使用 CN470-0 频段为例进行修改：

1、输入指令：sudo vim /etc/chirpstack/chirpstack.toml 打开需要修改参数的文件，修改内容如下：

```
enabled_regions = [
  "as923",
  "as923_2",
  "as923_3",
  "as923_4",
  "au915_0",
  "cn470_0",
  "cn470_10",
  "cn779",
  "eu433",
  "eu868",
  "in865",
  "ism2400",
  "kr920",
  "ru864",
  "us915_0",
  "us915_1",
]
```

2、输入指令：sudo vim /etc/chirpstack-gateway-bridge/chirpstack-gateway-bridge.toml 打开需要修改参数的文件，修改内容如下：

```
# Event topic template.
event_topic_template="cn470_0/gateway/{{ .GatewayID }}/event/{{ .EventType }}"

# Command topic template.
command_topic_template="cn470_0/gateway/{{ .GatewayID }}/command/#"

# MQTT authentication.
```

修改完成后输入指令：

sudo systemctl restart chirpstack-gateway-bridge

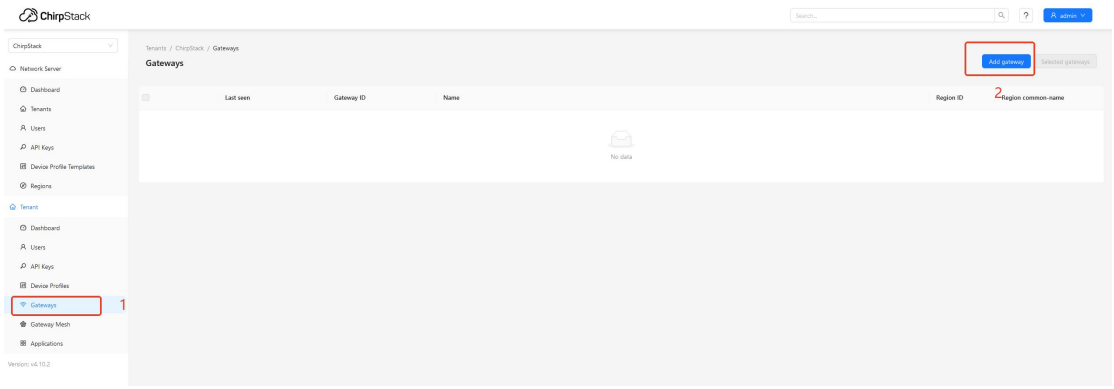
sudo systemctl restart chirpstack

注：这段配置表示 LoRaWAN 网络服务器被设置为在 CH470 频段工作，并且启用了上行信道 0，信道只设置了 0 意味着网络服务器将只监听这个信道上的上行消息。如需使用多个信道，参考 chirpstack 官方文档修改。

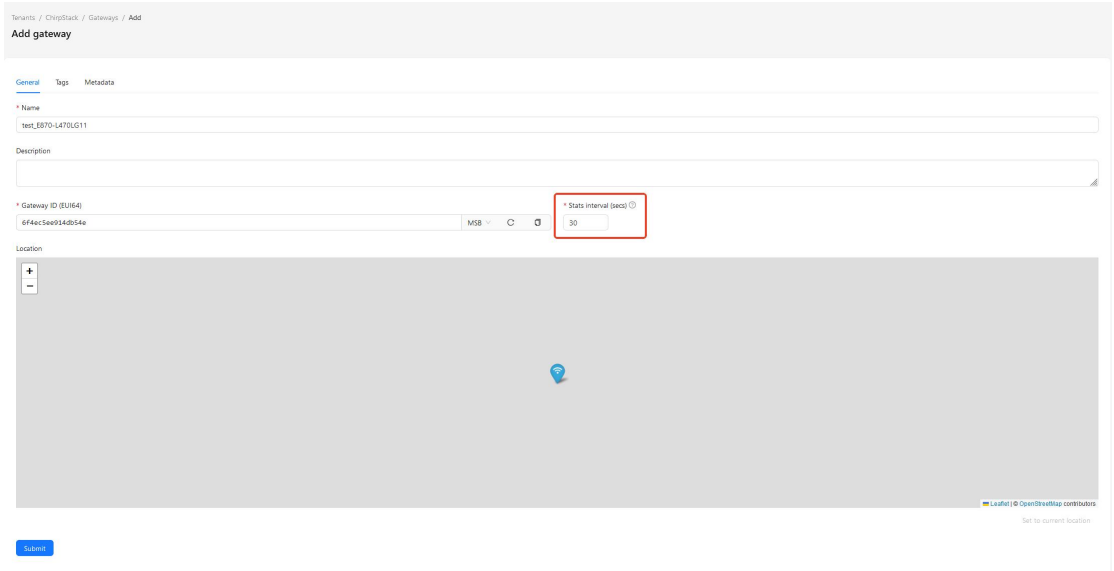
2、生成网关和节点

网关生成：

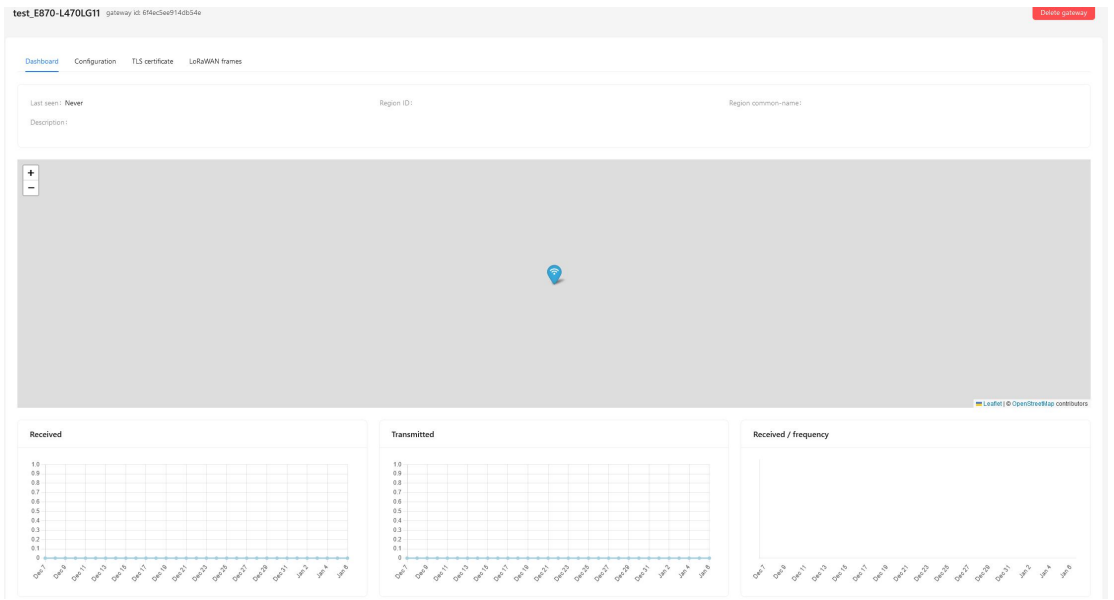
点击左侧栏中 Gateways，然后点击右上角 Add gateway 新建一个网关种类。



网关类型为 E870，填写网关 ID 也可以自动生成 ID，同个服务器不能使用相同的网关 ID。红框为网关的状态信息上报周期，E870 和网关均使用默认的 30 秒。点击 submit 创建网关。

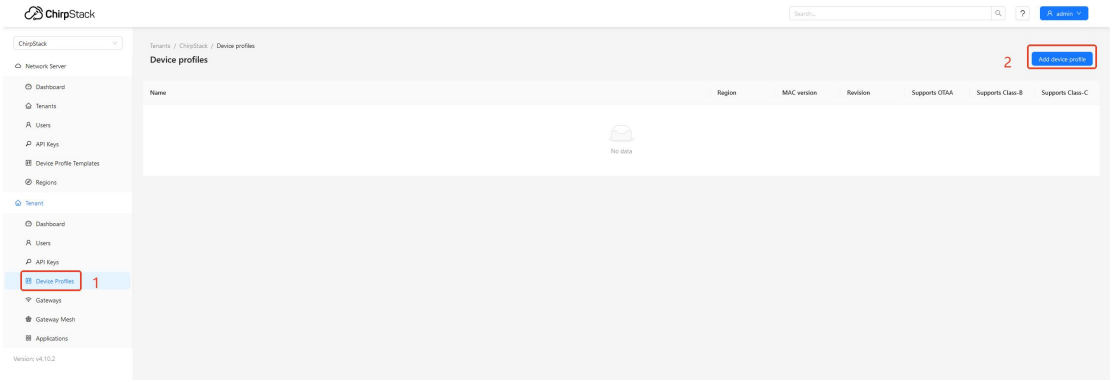


网关界面见下：

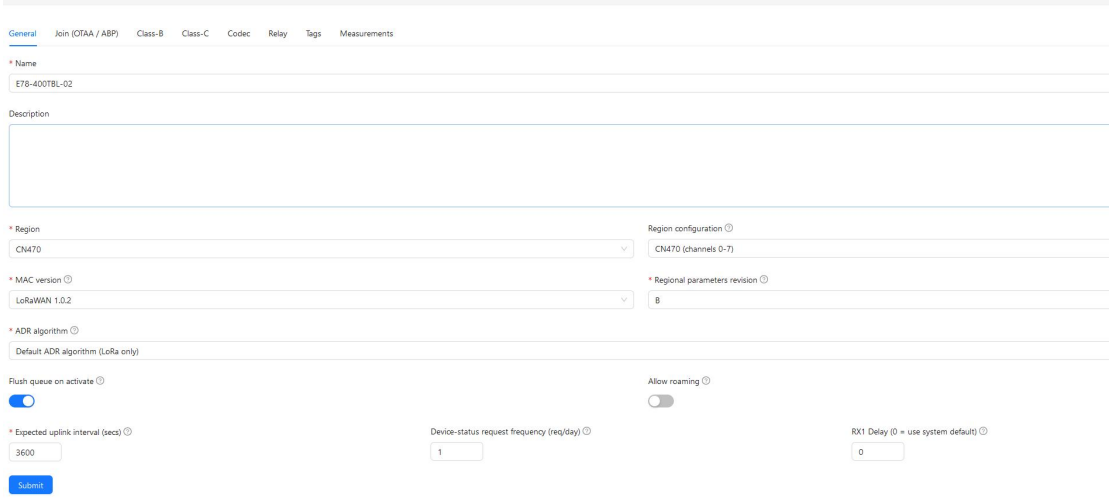


节点生成：

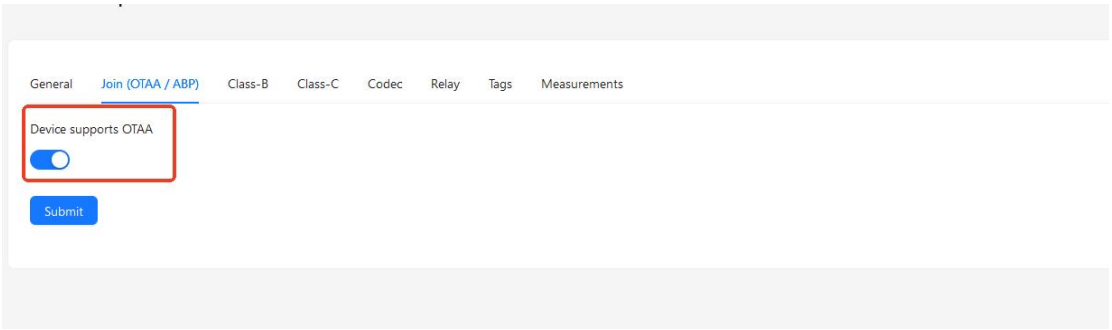
点击左侧框中 Device-profile，点击右上角“Adddevice profiles”新建一个节点种类。

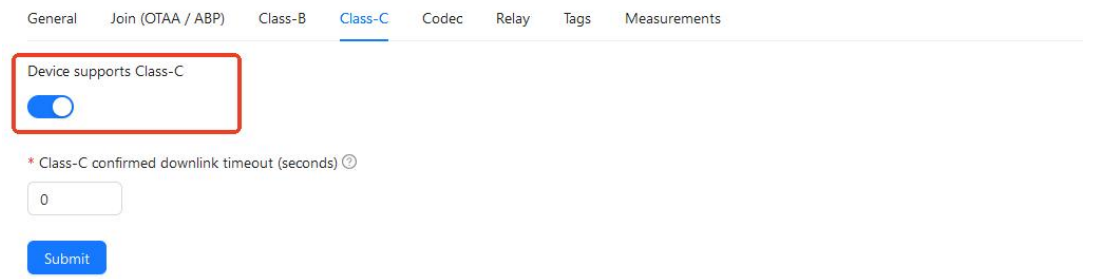


点击 Adddevice profiles 后填写的参数应和节点本身的频段和 LoRaWAN 版本一致，E78-400TBL-02 使用的参数信息见下图：



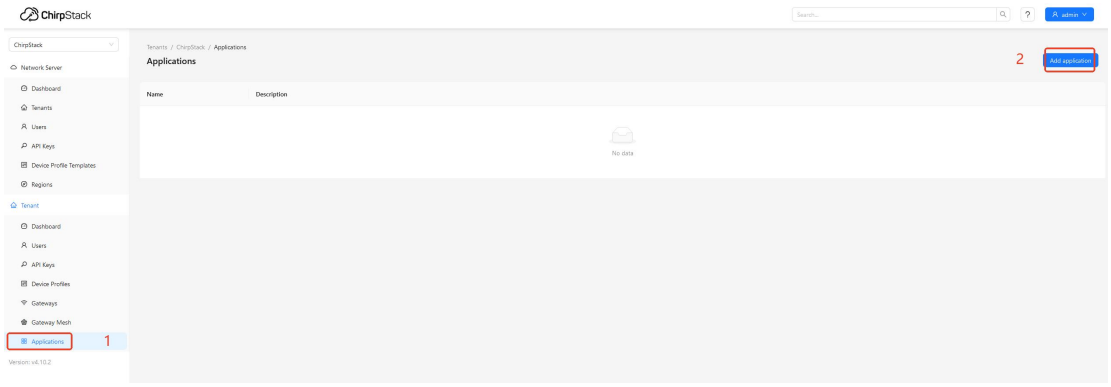
除去填写该信息还需打开 OTTA 模式并开启 CLASS C（OTTA 和 CLASS C 模式的具体基本知识请客户自行了解）：



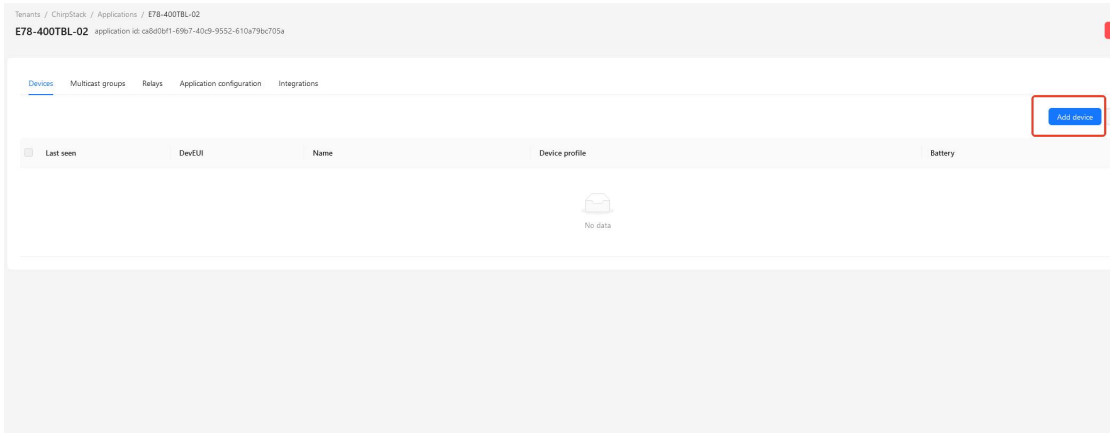


完成上述操作后点击 Submit 创建 deviceProfiles。

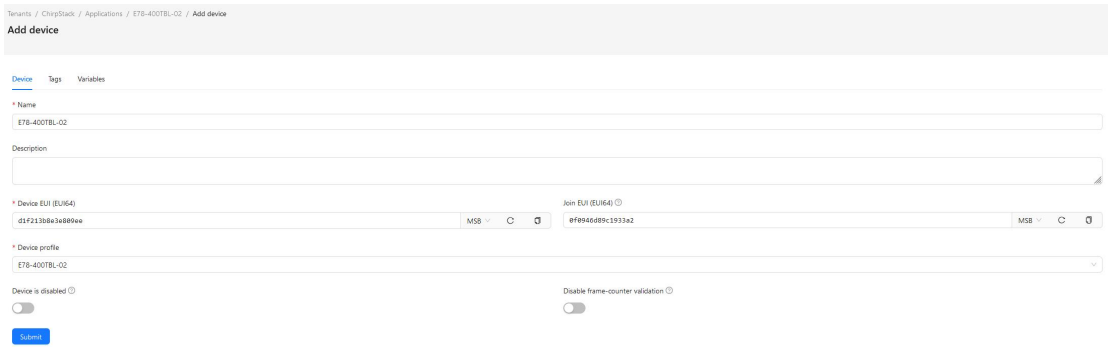
然后再点击左侧栏中 Applications，点击 Add application 新建一个应用，命名 E78-400TBL-02。



点击 Add device 生成一个节点：



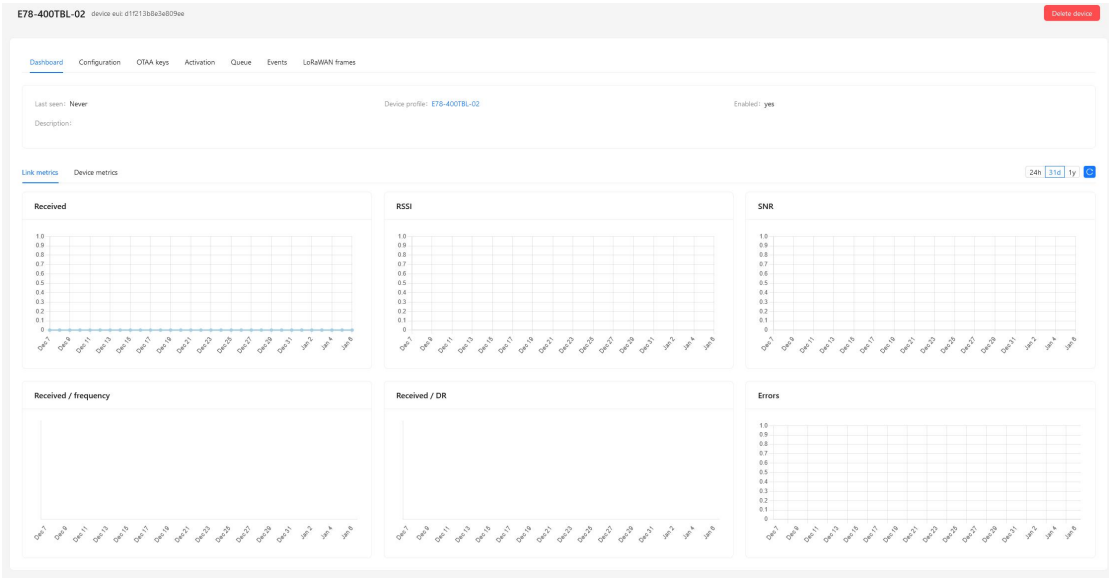
配置项见下图：



点击 submit 建立节点，随后会弹出添加 APPKEY 的界面，继续选择随机生成：

最后点击 submit 建立节点成功。

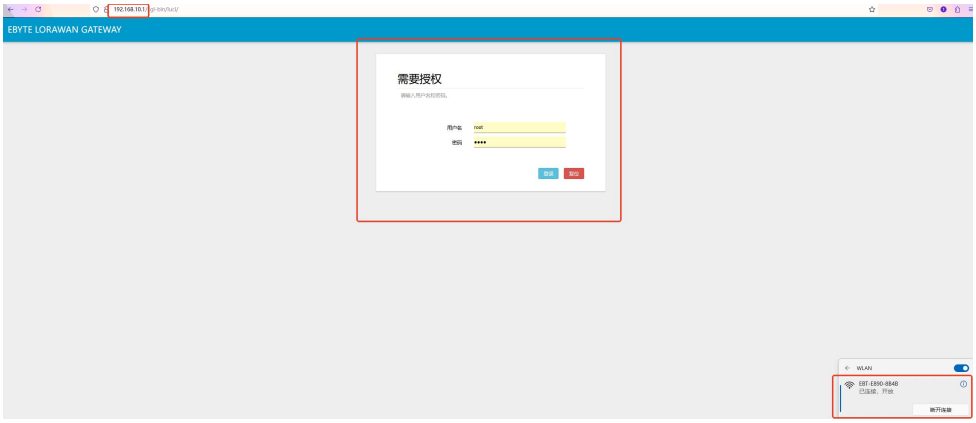
节点界面见下：

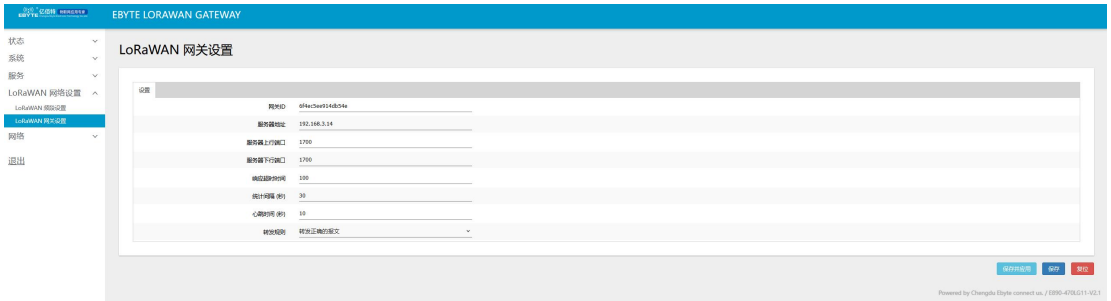


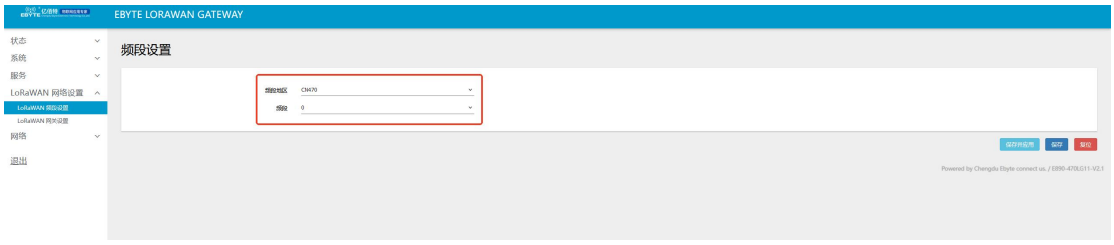
3、设备接入

网关设备接入：

打开 WiFi，连接网关的 WiFi，名称为 EBT-E890-XXXX。浏览器输入 192.168.10.1 进入配置页面。密码为 root。上边网关 ID 生成的参数为：，并修改 IP 地址为服务器的 IP 地址。



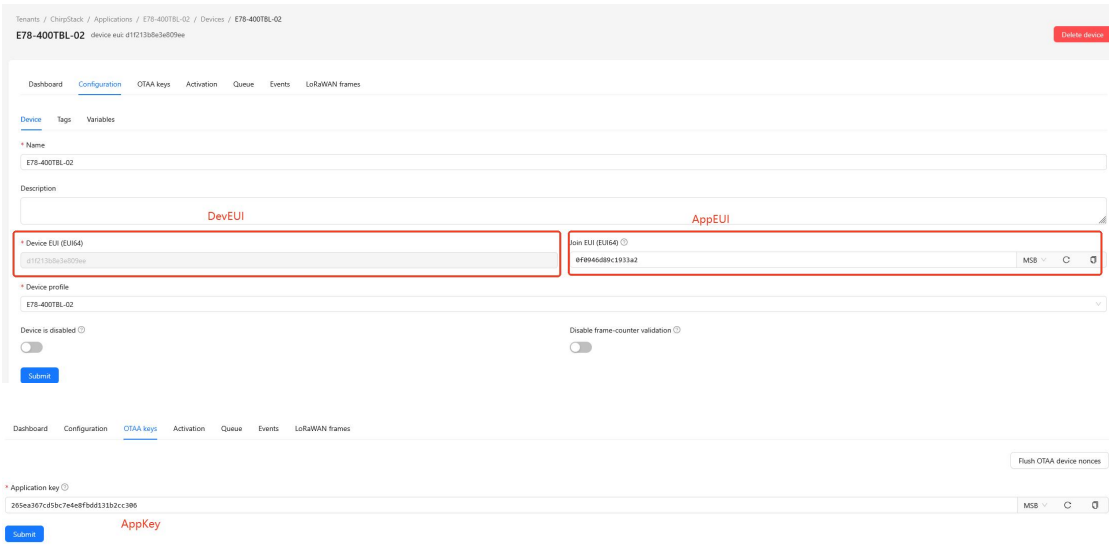


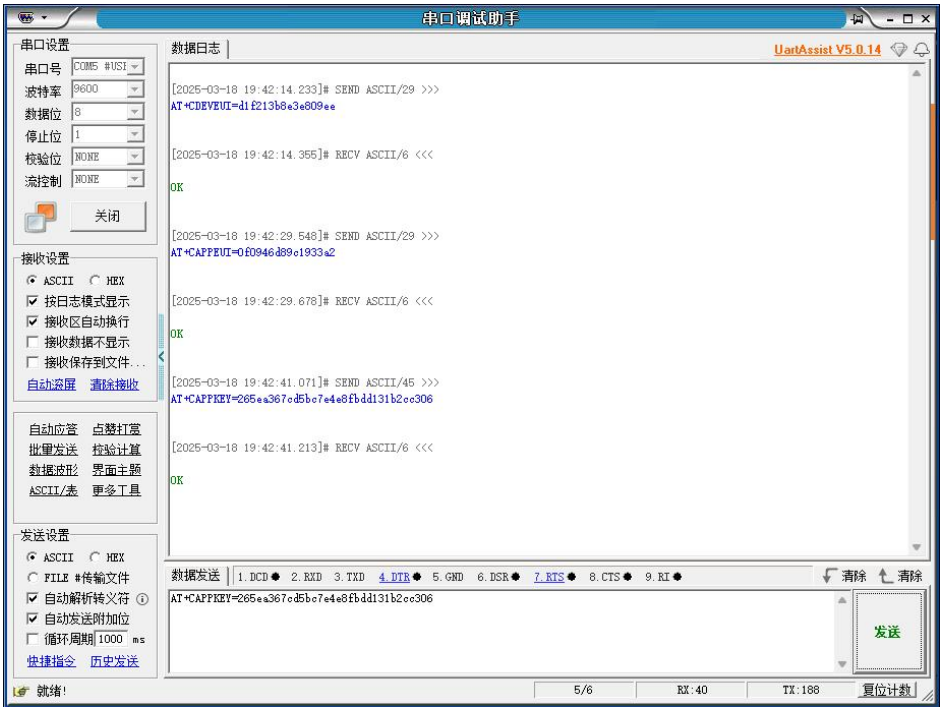


点击保存并应用，即可看到网关 LINK 灯亮起，即表示网关已连上服务器，在服务器中也可以看到网关状态显示网关 Online。

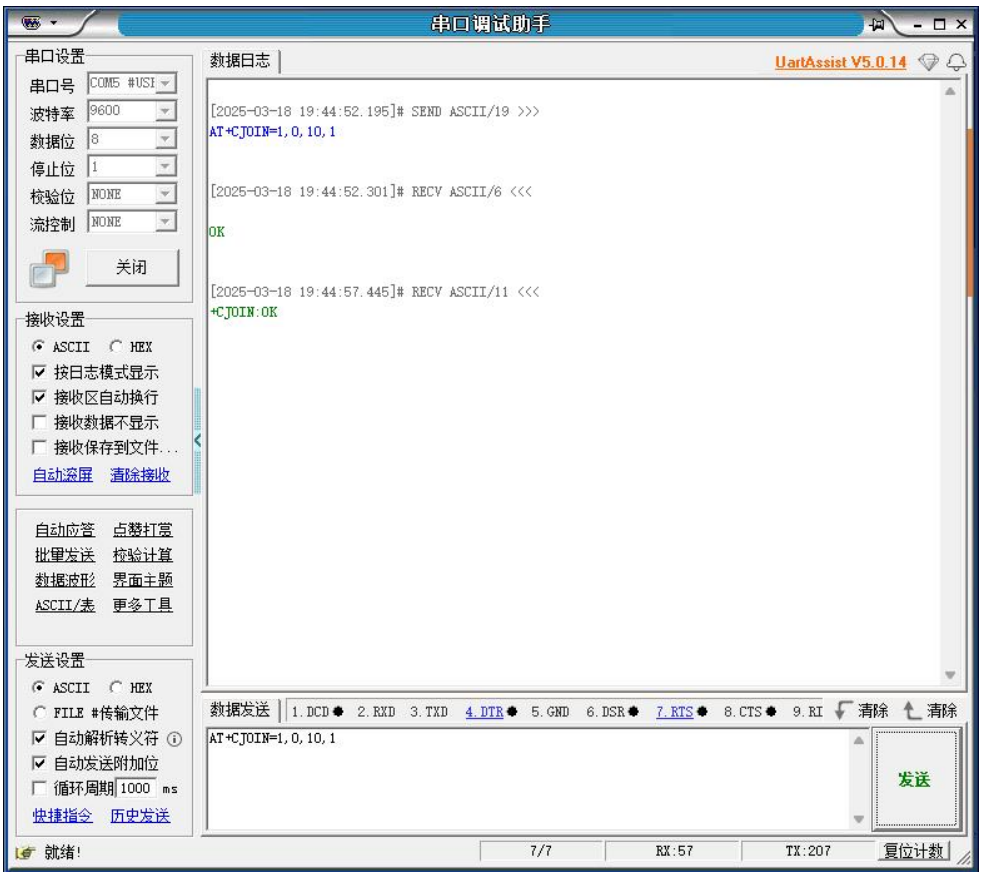
节点设备接入：

E78 节点设备通过 USB 连接电脑后，打开 chirpstack 服务器网页，找到配置好的节点参数；打开串口助手，通过 AT 指令配置 AppEUI，DevEUI 以及 AppKey 参数。

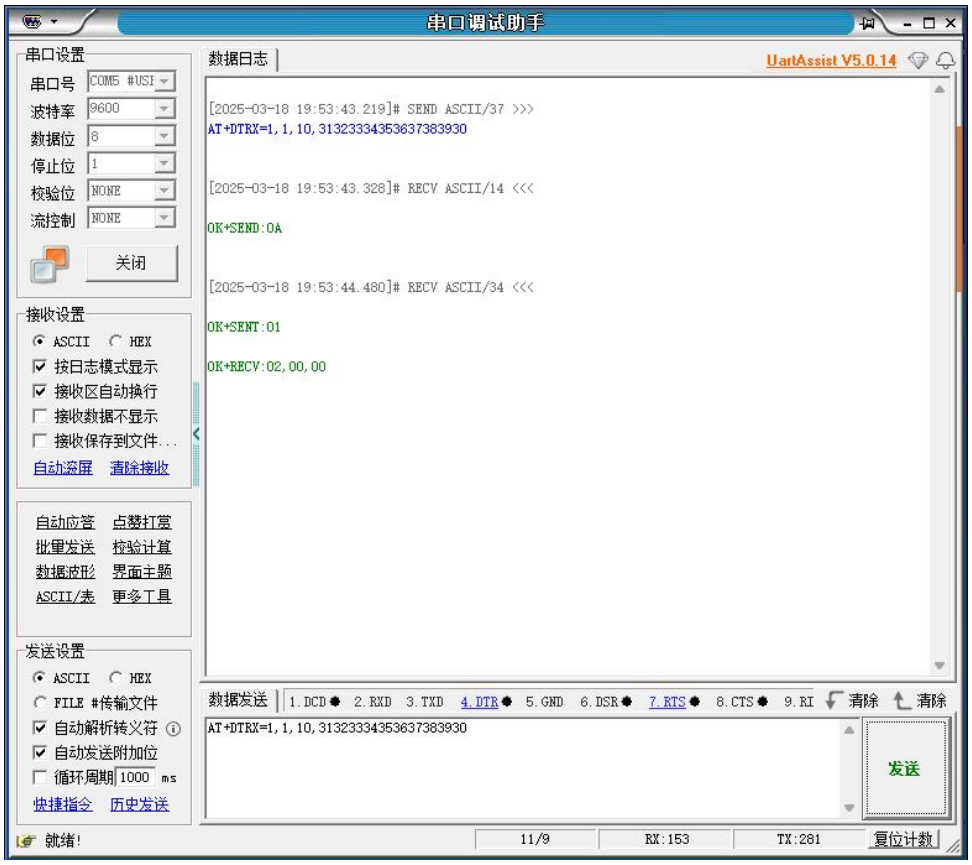


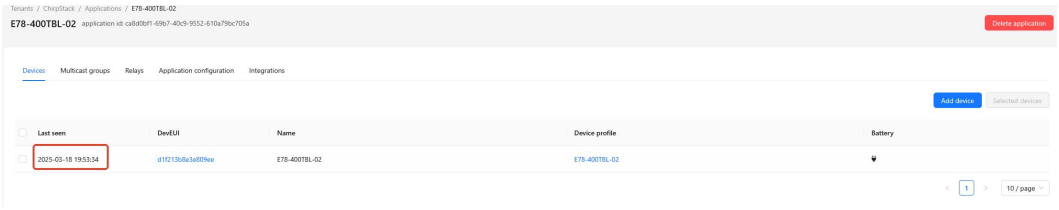


当 E78-470LN22S 参数配置完成后，发送入网指令 AT+CJOIN=1,0,10,1，回复+CJOIN:OK，即入网成功。



此时可通过指令“AT+DTRX=1,1,10,31323334353637383930”发送数据测试通信是否正常，如返回下图结果则通信正常，且服务器中显示最后一次通信时间。





到此已完成 LoRaWAN 节点模块和 LoRaWAN 网关接入 Chirpstack 服务器建立 LoRaWAN 网络。