Connecting Dragino LPS8 indoor Gateway to Noria Chirpstack server

First log in the system

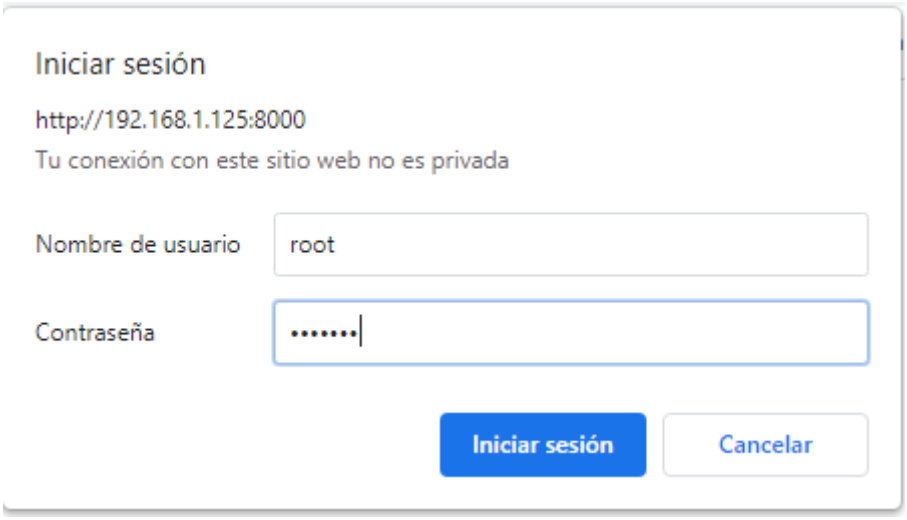
Since it has DHCP enabled by default, let's connect to our router thru a patchcord.

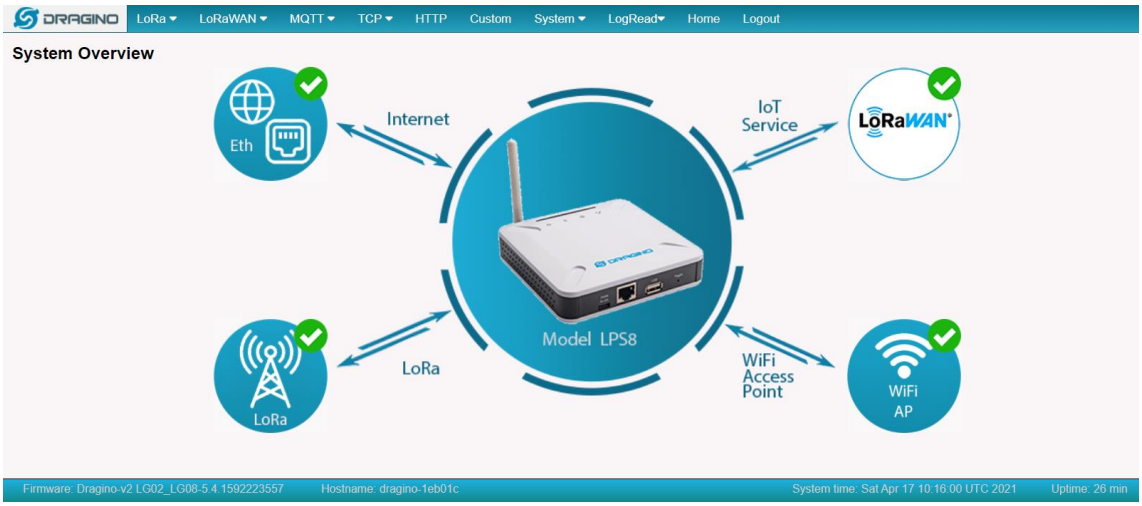
Look on the router ip map the corresponding IP adress

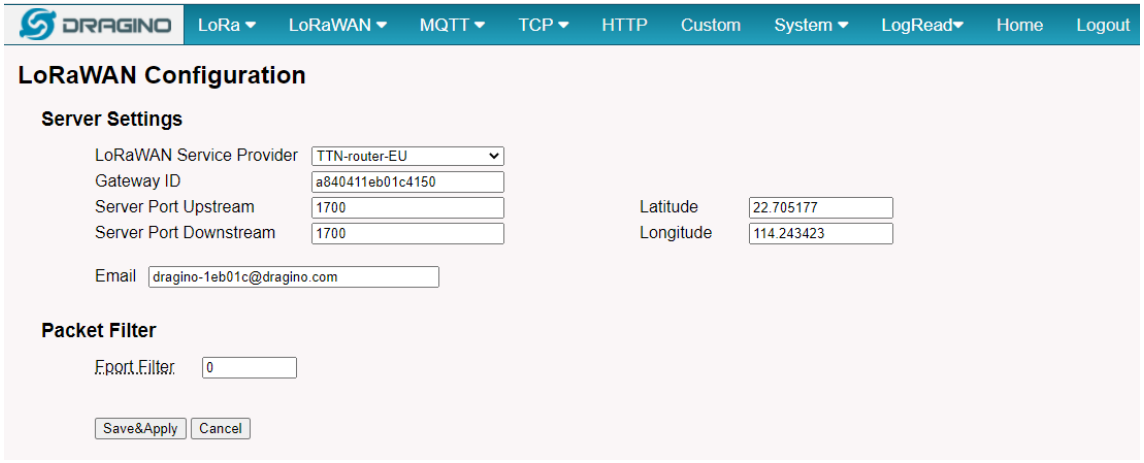
This time it is 192.168.1.125

Let's enter on the Gateway 192.168.1.125:8000

Root





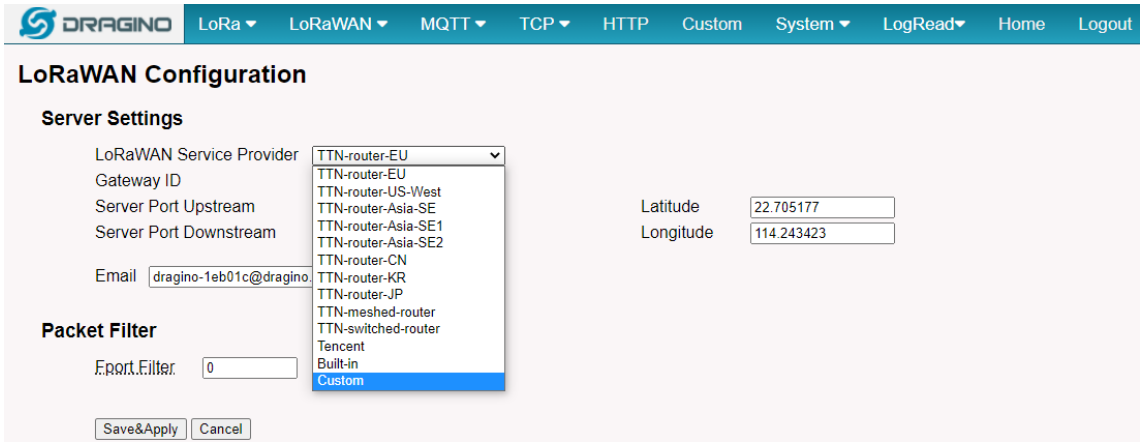


These are the user setting to connect to TTN

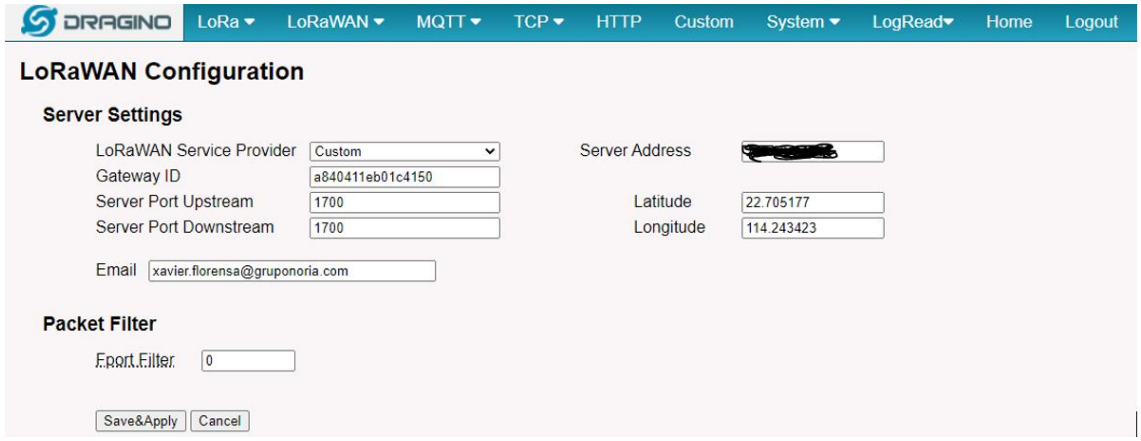
But we want to connect to Noria chirpstack server

So let's change these settings

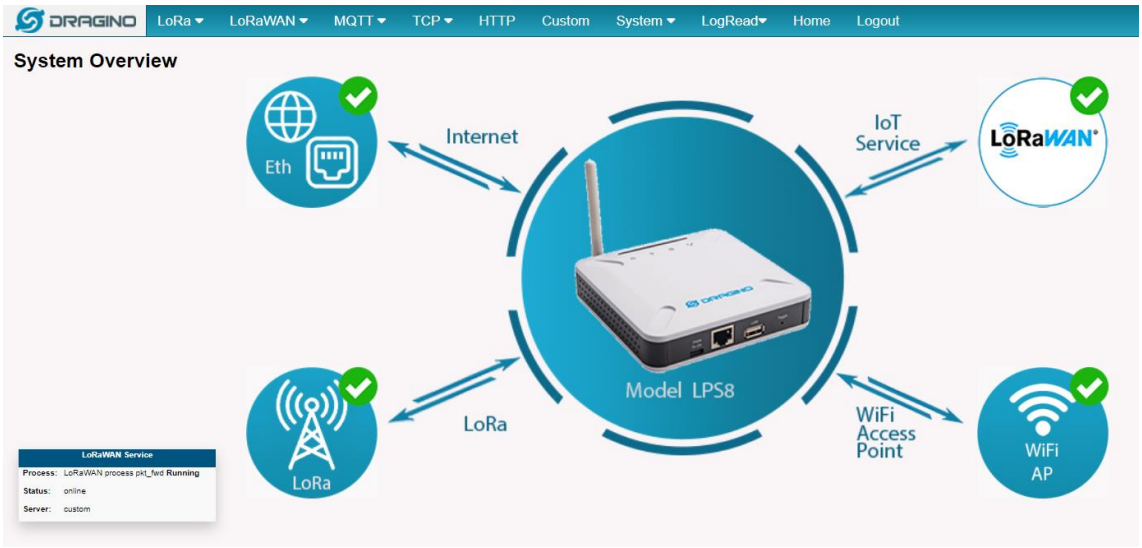
We have to select custom

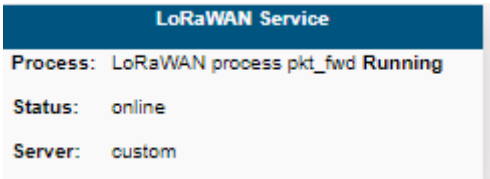


And introduce the Noria Chirpstack server address



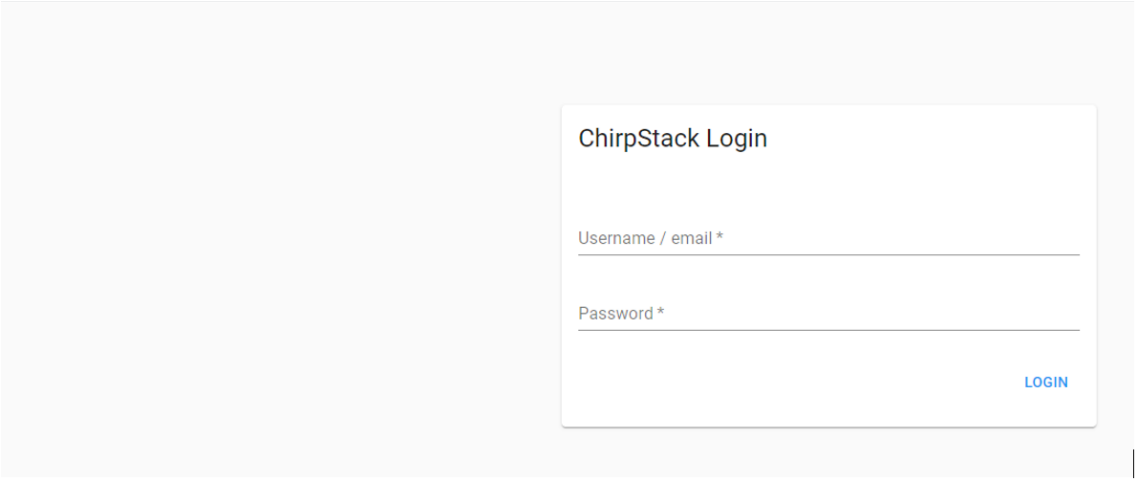
Save and Apply



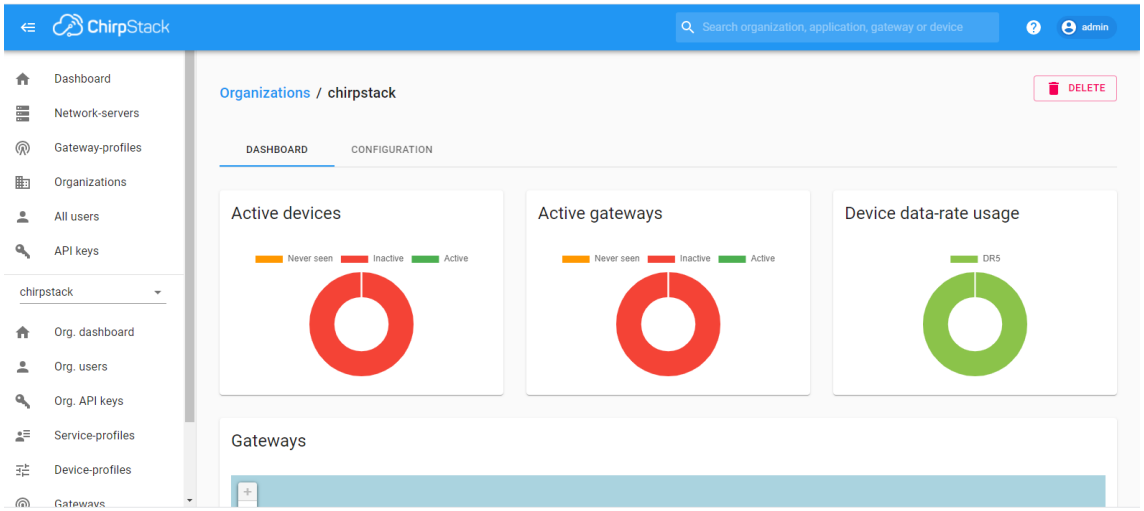


Now Let’s jump to The Noria Network server (Chirpstack)

Serverip address:8080

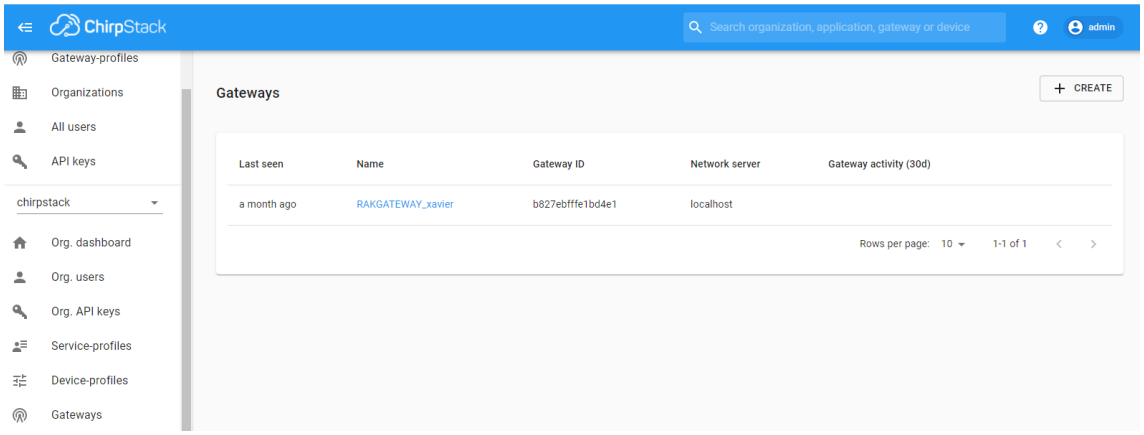


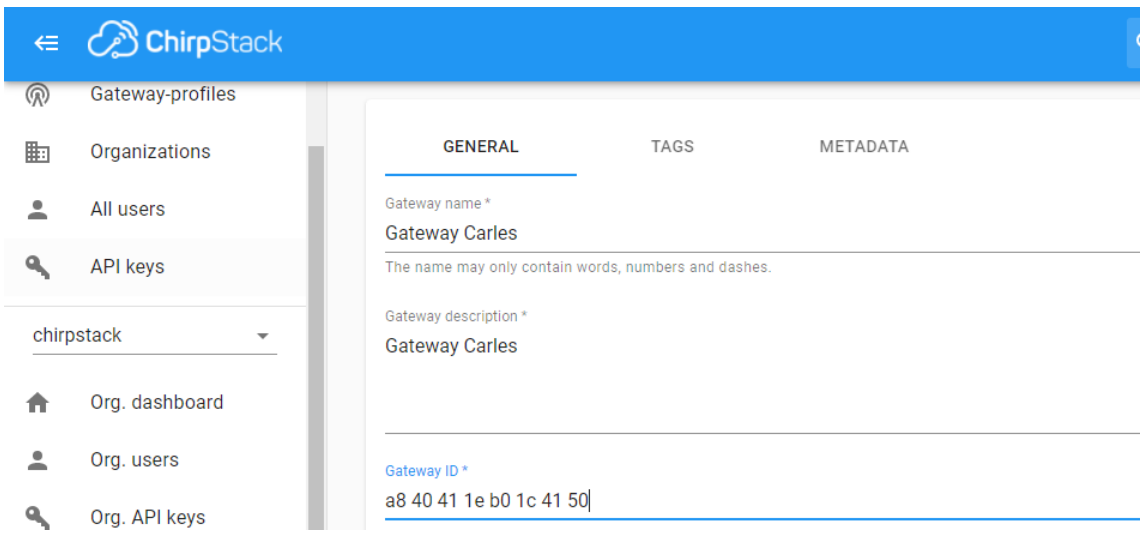
Here we are

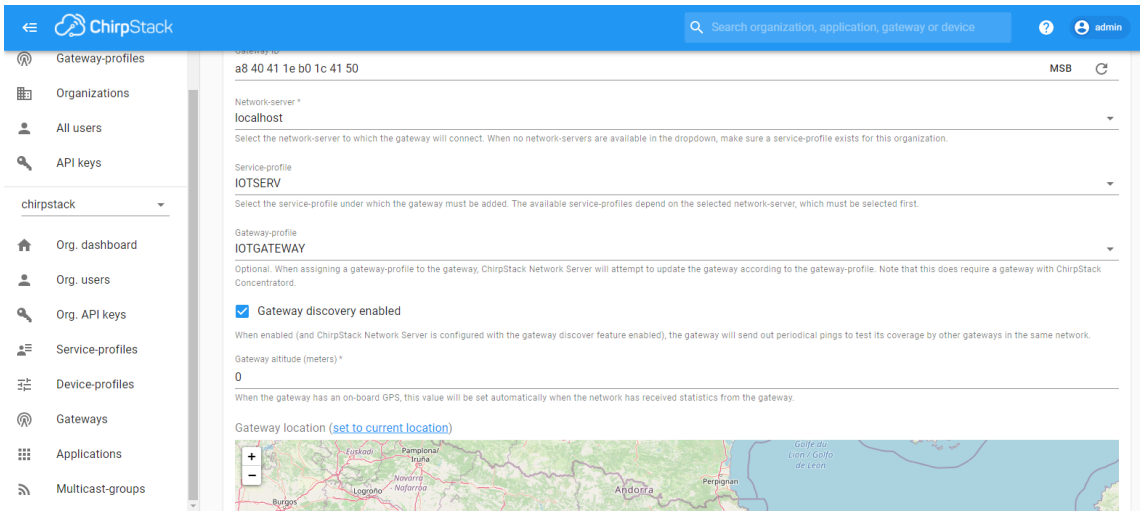


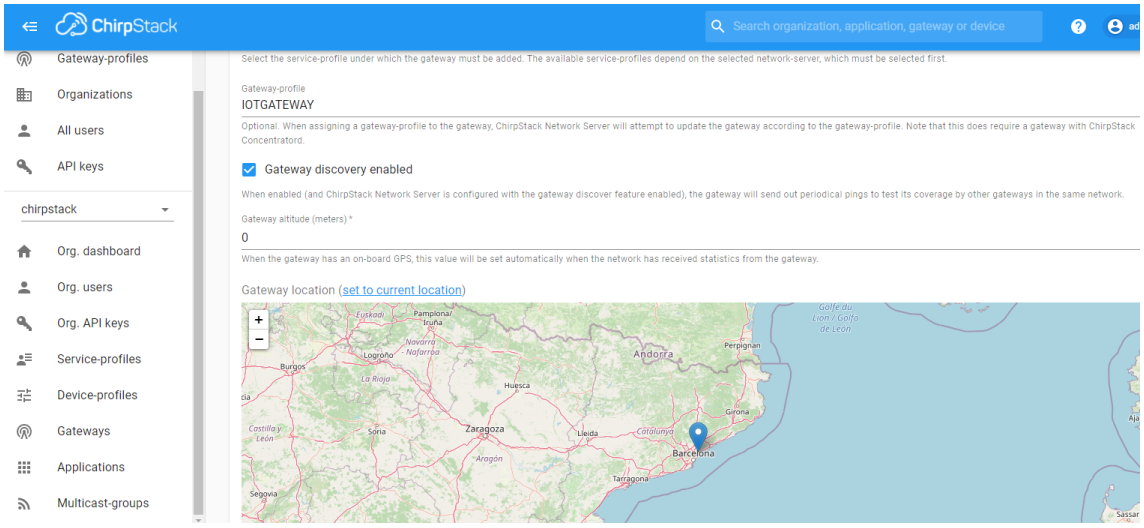
Now let’s settle the new Gateway

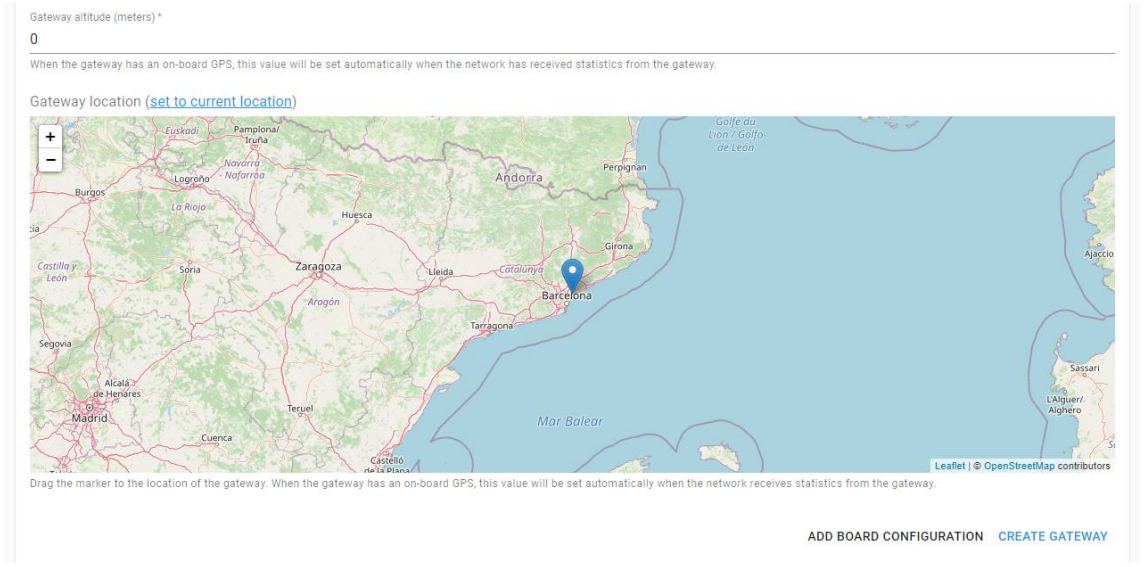
Create





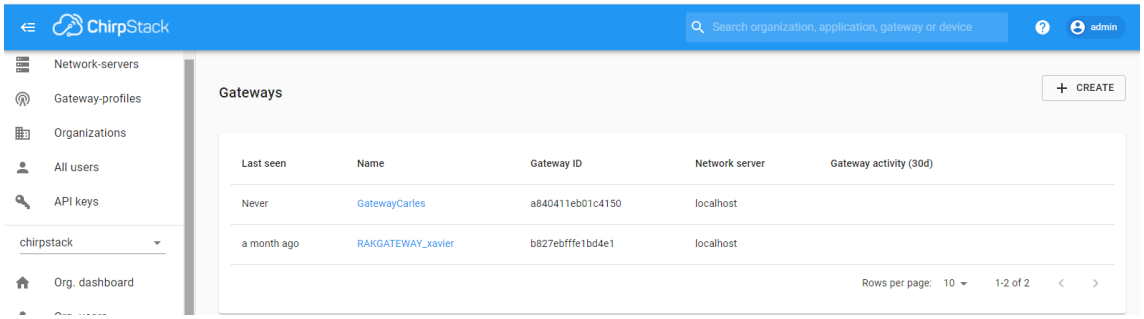






And create Gateway

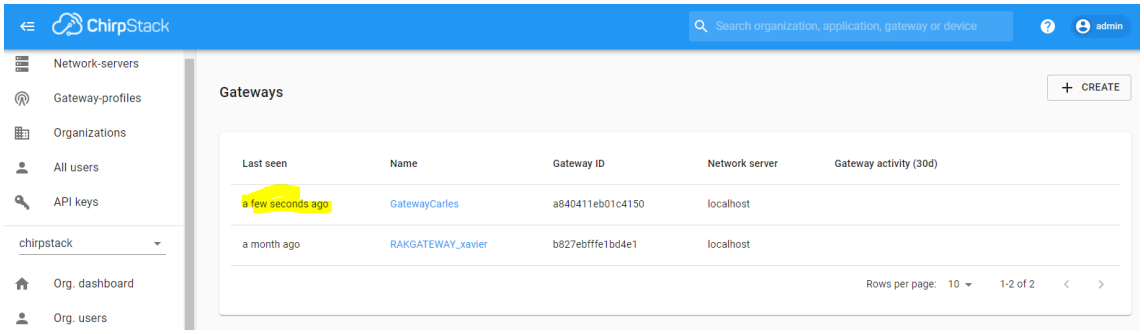
The Gateway is there

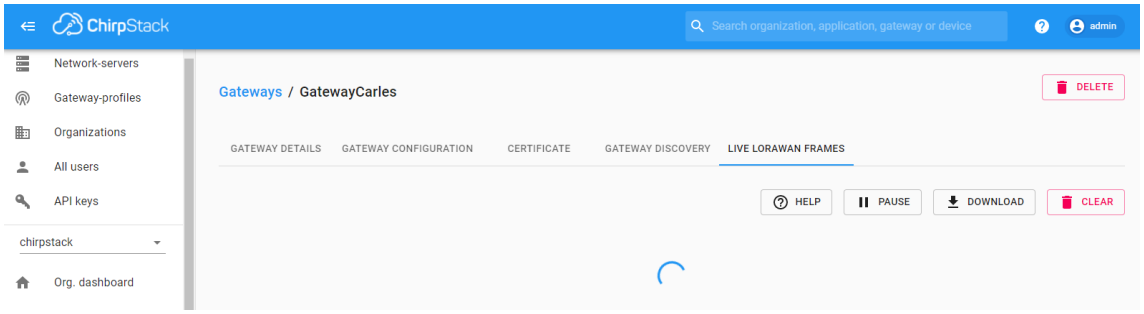


But appears as not connected

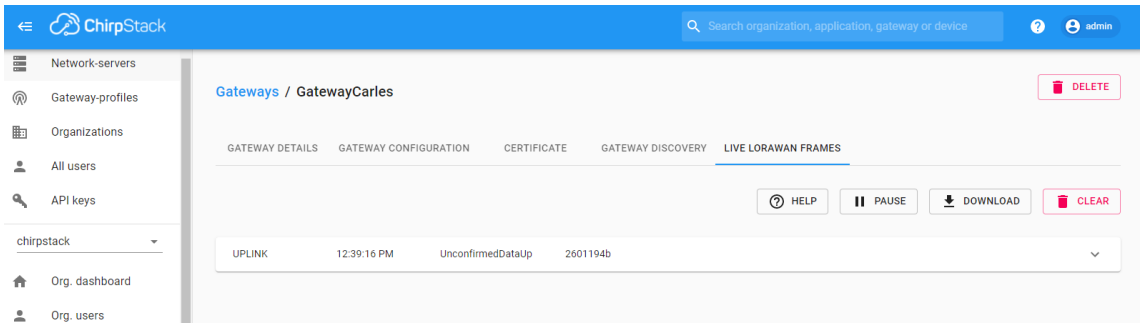
Maybe it takes some minutes

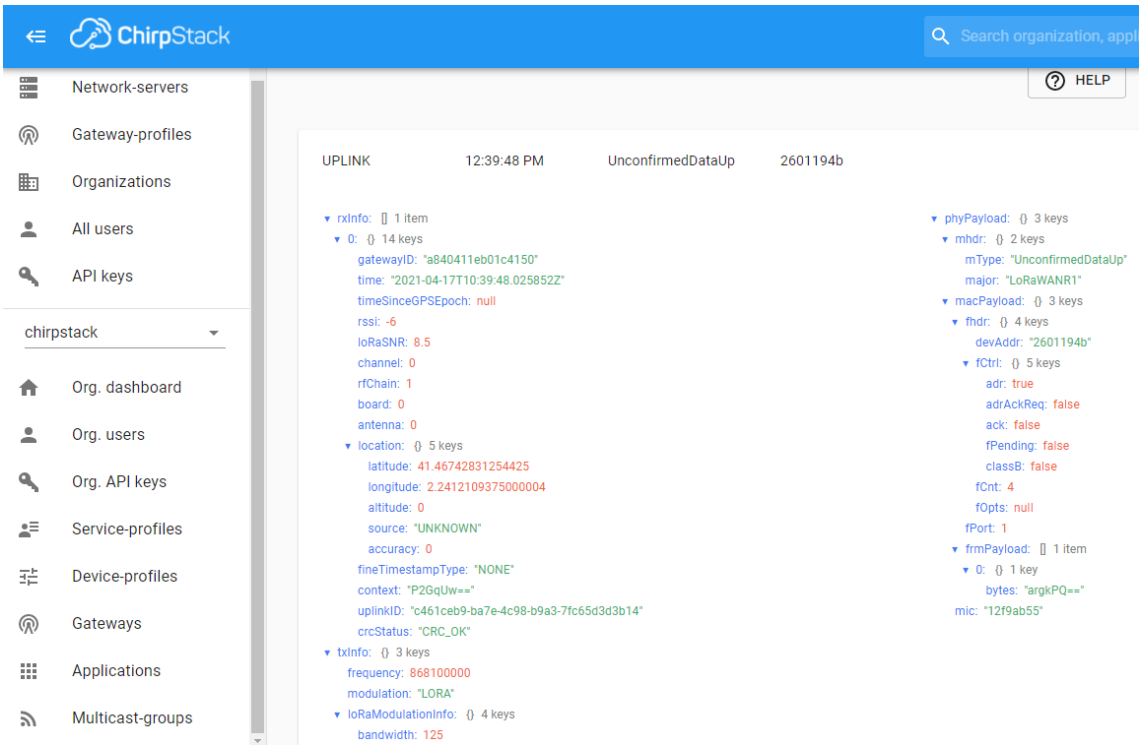
Let’s connect a sensor





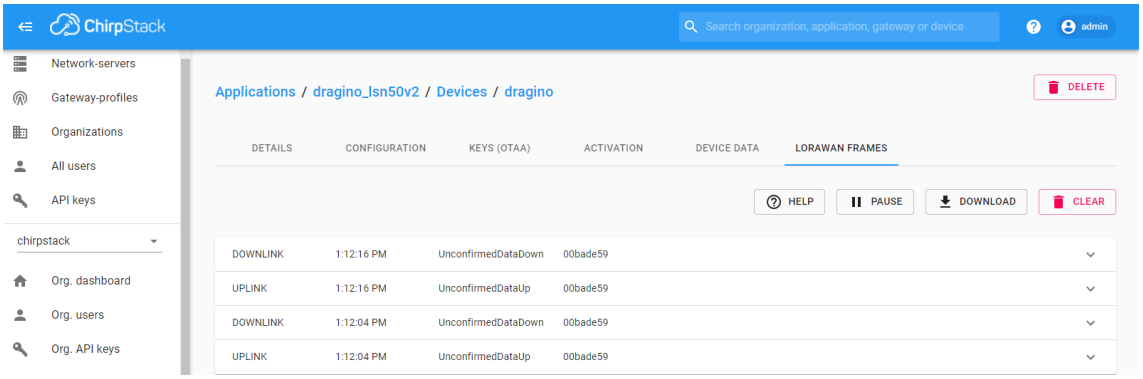
After 40 seconds we have the first uplink





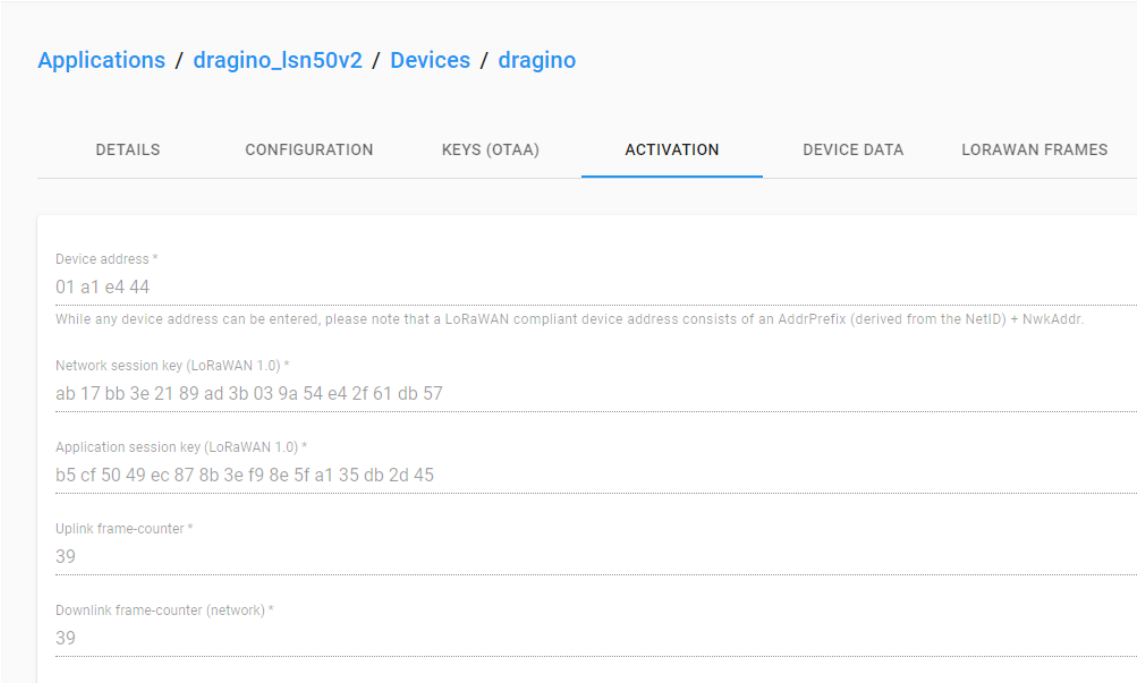
The application is not correctly set for ABP since we do not see any traffic on the application

Let’s try to setup a OTAA application with a Dragino LSN50 v2

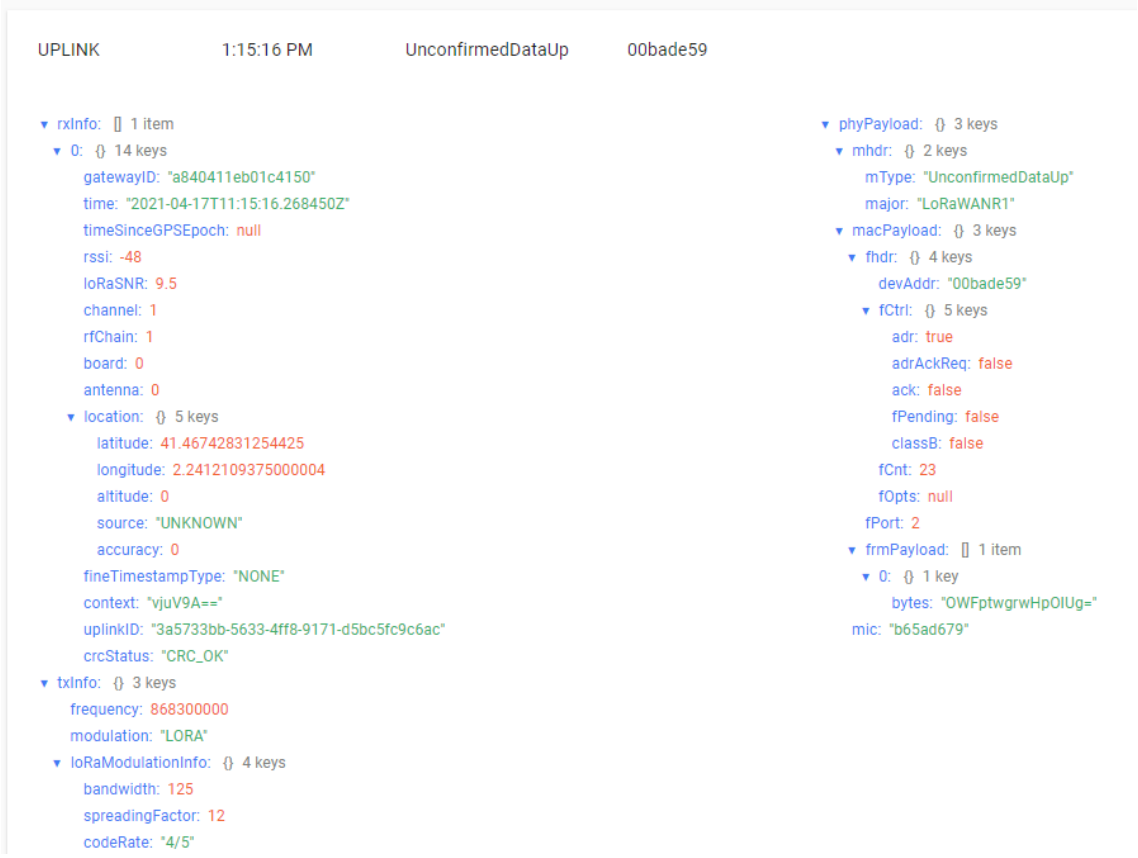


We have only entered the App key

After first join request, teh device is already on the system.

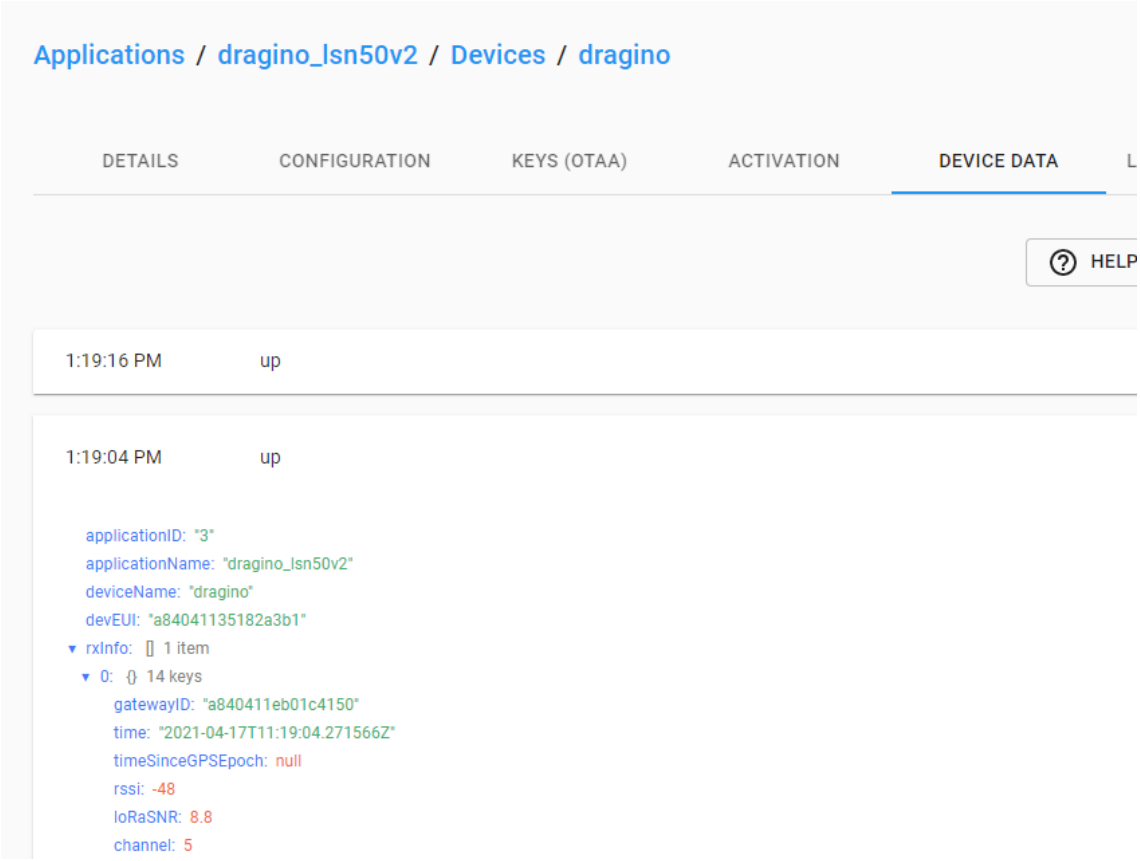


The device address changes on each join request



There is no data

Since we have to look at the data tab

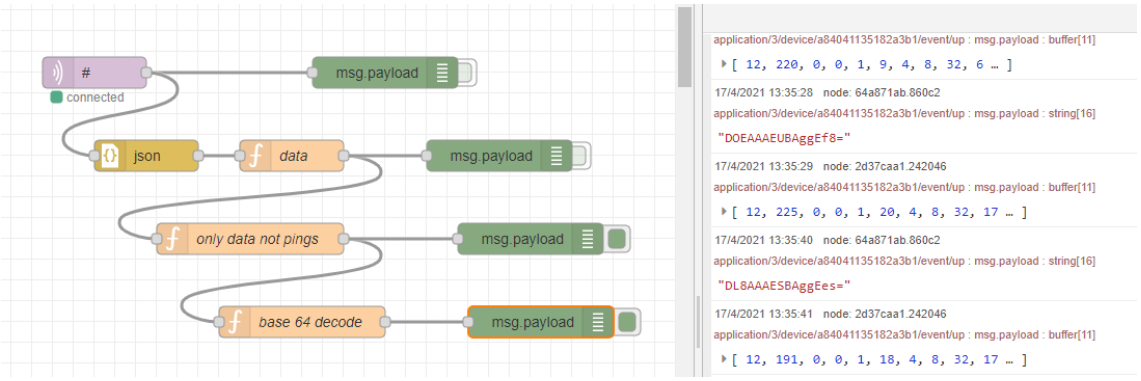

Applications / dragino_lsn50v2 / Devices / dragino
DETAILS
CONFIGURATION
KEYS (OTAA)
ACTIVATION
DEVICE DATA
HELP
1:19:16 PM up
1:19:04 PM up
applicationID: "3"
applicationName: "dragino_lsn50v2"
deviceName: "dragino"
devEUI: "a84041135182a3b1"
rxInfo: [] 1 item
0: {} 14 keys
gatewayID: "a840411eb01c4150"
time: "2021-04-17T11:19:04.271566Z"
timeSinceGPSEpoch: null
rssi: -48
loRaSNR: 8.8
channel: 5

```
        source: "UNKNOWN"
        accuracy: 0
      fineTimestampType: "NONE"
      context: "y9KqPA=="
      uplinkID: "7d126556-3634-44ea-9041-b288a0aa4c1b"
      crcStatus: "CRC_OK"
  ▾ txInfo: {} 3 keys
    frequency: 867500000
    modulation: "LORA"
  ▾ loRaModulationInfo: {} 4 keys
      bandwidth: 125
      spreadingFactor: 12
      codeRate: "4/5"
      polarizationInversion: false
  adr: true
  dr: 0
  fCnt: 42
  fPort: 2
  data: "DNwAAAERBATECFo="
  objectJSON: ""
  tags: {} 0 keys
  confirmedUplink: false
  devAddr: "00bade59"
```


This data is Base64 encoded

DNwAAAERBATECFo=

But we can decode the payload



"DM4AAAERBALkCgI="

17/4/2021 13:45:05 node: 2d37caa1.242046

application/3/device/a84041135182a3b1/event/up : msg.payload : buffer[11]

▾ *buffer[11]* raw

▾ [0 … 9]

0: 0xc

1: 0xce

2: 0x0

3: 0x0

4: 0x1

5: 0x11

6: 0x4

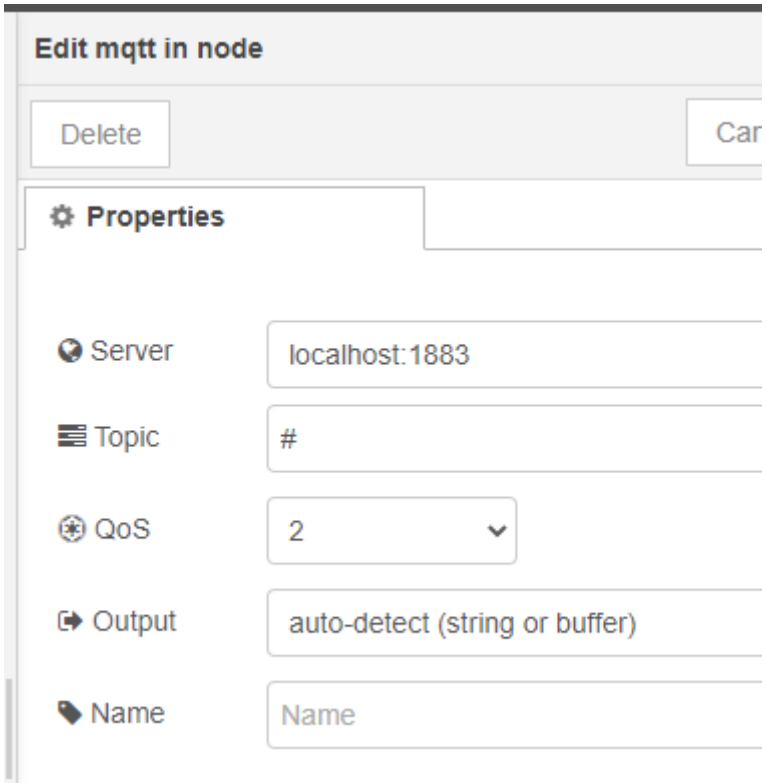
7: 0x2

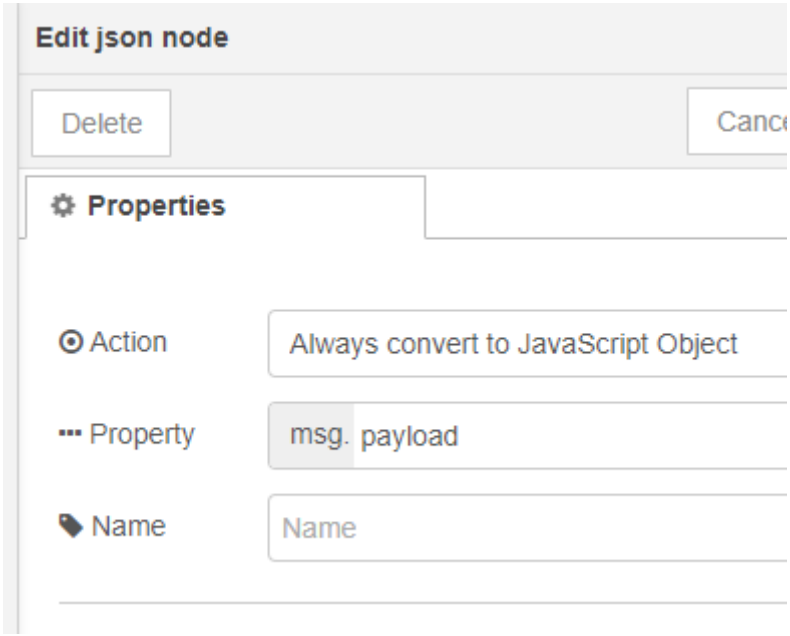
8: 0xe4

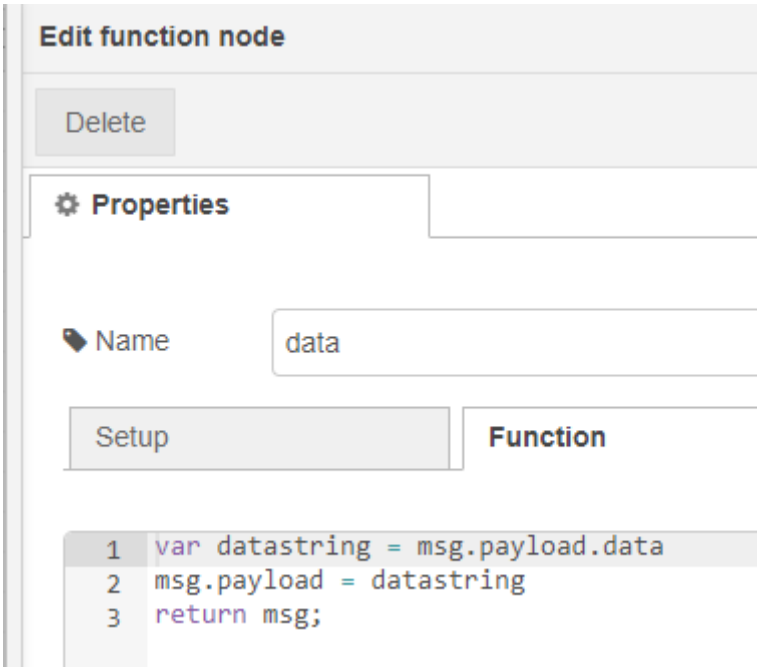
9: 0xa

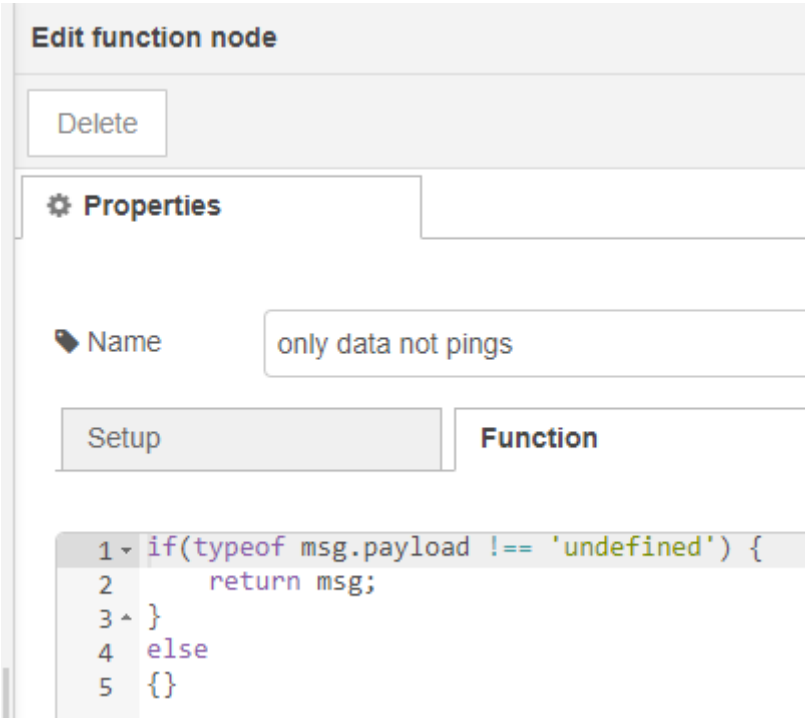
▾ [10 … 10]

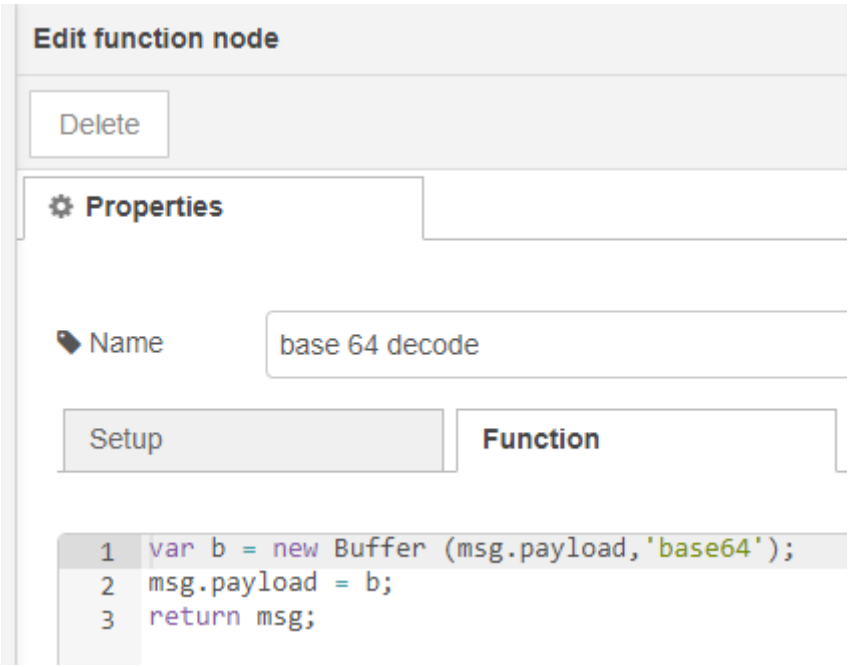
10: 0x2




Edit json node
Delete
Cance
Properties
Action
Always convert to JavaScript Object
Property
msg. payload
Name
Name


Edit function node
Delete
Properties
Name
data
Setup
Function
1 var datastring = msg.payload.data
2 msg.payload = datastring
3 return msg;



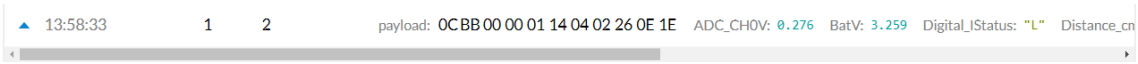


The Device Address changes on each join request to Chirpstack

So if we have a device sending to a chirpstack Gateway and then we also plug a TTN Gateway, the device will no longer be visible from the TTN application since it has changed the device address.

To be visible from the TTN application we have to stop the Chirpstack Gateway

Let's compare to a typical payload on TTN



So payload are 11 bytes

And same we have received on Chirpstack

Starting with 0C

Now let’s try to decode the payload with node-red

We try to copy the payload decoder used in TTN

function Decoder(bytes, port) {

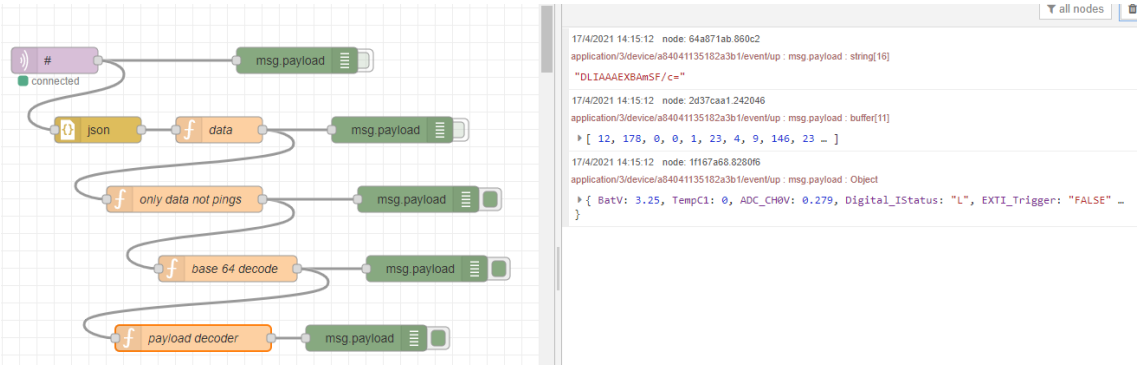
```
var mode=(bytes[6] & 0x7C)>>2;
var decode = {};
if(mode!=2)
{
  decode.BatV=(bytes[0]<<8 | bytes[1])/1000;
  decode.TempC1= parseFloat(((bytes[2]<<24>>16 | bytes[3])/10).toFixed(2));
  decode.ADC_CH0V=(bytes[4]<<8 | bytes[5])/1000;
  decode.Digital_IStatus=(bytes[6] & 0x02)? "H":"L";
  if(mode!=6)
  {
            decode.EXTI_Trigger=(bytes[6] & 0x01)? "TRUE":"FALSE";
  decode.Door_status=(bytes[6] & 0x80)? "CLOSE":"OPEN";
  }
}

if(mode=='0')
{
  decode.Work_mode="IIC";
  if((bytes[9]<<8 | bytes[10])===0)
  {
    decode.Illum=(bytes[7]<<24>>16 | bytes[8]);
  }
  else
  {
  decode.TempC_SHT=parseFloat(((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
  decode.Hum_SHT=parseFloat(((bytes[9]<<8 | bytes[10])/10).toFixed(1));
  }
}
else if(mode=='1')
{
  decode.Work_mode=" Distance";
```

```
  decode.Distance_cm=parseFloat(((bytes[7]<<8 | bytes[8])/10) .toFixed(1));
  if((bytes[9]<<8 | bytes[10])!=65535)
  {
    decode.Distance_signal_strength=parseFloat((bytes[9]<<8 | bytes[10]) .toFixed(0));
  }
}
else if(mode=='2')
{
  decode.Work_mode=" 3ADC";
  decode.BatV=bytes[11]/10;
  decode.ADC_CH0V=(bytes[0]<<8 | bytes[1])/1000;
  decode.ADC_CH1V=(bytes[2]<<8 | bytes[3])/1000;
  decode.ADC_CH4V=(bytes[4]<<8 | bytes[5])/1000;
  decode.Digital_IStatus=(bytes[6] & 0x02)? "H":"L";
  decode.EXTI_Trigger=(bytes[6] & 0x01)? "TRUE":"FALSE";
  decode.Door_status=(bytes[6] & 0x80)? "CLOSE":"OPEN";
  if((bytes[9]<<8 | bytes[10])===0)
  {
    decode.Illum=(bytes[7]<<24>>16 | bytes[8]);
  }
  else
  {
  decode.TempC_SHT=parseFloat(((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
  decode.Hum_SHT=parseFloat(((bytes[9]<<8 | bytes[10])/10) .toFixed(1));
  }
}
else if(mode=='3')
{
  decode.Work_mode="3DS18B20";
  decode.TempC2=parseFloat(((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
  decode.TempC3=parseFloat(((bytes[9]<<8 | bytes[10])/10) .toFixed(1));
```

```
}
else if(mode=='4')
{
  decode.Work_mode="Weight";
  decode.Weight=(bytes[7]<<24>>16 | bytes[8]);
}
else if(mode=='5')
{
  decode.Work_mode="Count";
  decode.Count=(bytes[7]<<24 | bytes[8]<<16 | bytes[9]<<8 | bytes[10]);
}

  if((bytes.length==11)||(bytes.length==12))
  {
  return decode;
  }
}
```



You can find the code here

https://github.com/xavierflorensa/Chirpstack-Gateway-to-Node-red-local/blob/master/node-red%20chirpstack%20payload%20decoder%20dragino%20lsn50v2.txt

Yes, with this decoder

```
object
  BatV: 3.246
  TempC1: 0
  ADC_CH0V: 0.269
  Digital_IStatus: "L"
  EXTI_Trigger: "FALSE"
  Door_status: "OPEN"
  Work_mode: " Distance"
  Distance_cm: 245
  Distance_signal_strength: 6146
```


Edit function node

Delete

Properties

Name: payload decoder

Setup | **Function** | Close

```
1  var bytes =msg.payload;
2
3  //function Decoder(bytes, port) {
4  var mode=(bytes[6] & 0x7C)>>2;
5  var decode = {};
6  if(mode!=2)
7  {
8    decode.BatV=(bytes[0]<<8 | bytes[1])/1000;
9    decode.TempC1= parseFloat(((bytes[2]<<24>>16 | bytes[3])/10).toFixed(2));
10   decode.ADC_CH0V=(bytes[4]<<8 | bytes[5])/1000;
11   decode.Digital_IStatus=(bytes[6] & 0x02)? "H":"L";
12   if(mode!=6)
13   {
14      decode.EXTI_Trigger=(bytes[6] & 0x01)? "TRUE":"FALSE";
15   decode.Door_status=(bytes[6] & 0x80)? "CLOSE":"OPEN";
16   }
17 }
18
19 if(mode=='0')
20 {
21   decode.Work_mode="IIC";
```

```
  if((bytes[9]<<8 | bytes[10])===0)
  {
    decode.Illum=(bytes[7]<<24>>16 | bytes[8]);
  }
  else
  {
  decode.TempC_SHT=parseFloat(((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
  decode.Hum_SHT=parseFloat(((bytes[9]<<8 | bytes[10])/10).toFixed(1));
  }
}
else if(mode=='1')
{
  decode.Work_mode=" Distance";
  decode.Distance_cm=parseFloat(((bytes[7]<<8 | bytes[8])/10) .toFixed(1));
  if((bytes[9]<<8 | bytes[10])!=65535)
  {
    decode.Distance_signal_strength=parseFloat((bytes[9]<<8 | bytes[10]) .toFixed(0));
  }
```

```
  }
}
else if(mode=='2')
{
  decode.Work_mode=" 3ADC";
  decode.BatV=bytes[11]/10;
  decode.ADC_CH0V=(bytes[0]<<8 | bytes[1])/1000;
  decode.ADC_CH1V=(bytes[2]<<8 | bytes[3])/1000;
  decode.ADC_CH4V=(bytes[4]<<8 | bytes[5])/1000;
  decode.Digital_IStatus=(bytes[6] & 0x02)? "H":"L";
  decode.EXTI_Trigger=(bytes[6] & 0x01)? "TRUE":"FALSE";
  decode.Door_status=(bytes[6] & 0x80)? "CLOSE":"OPEN";
  if((bytes[9]<<8 | bytes[10])===0)
  {
    decode.Illum=(bytes[7]<<24>>16 | bytes[8]);
  }
  else
  {
  decode.TempC_SHT=parseFloat(((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
  decode.Hum_SHT=parseFloat(((bytes[9]<<8 | bytes[10])/10) .toFixed(1));
```

```
  decode.Hum_SHT=parseFloat(((bytes[9]<<8 | bytes[10])/10) .toFixed(1));
  }
}
else if(mode=='3')
{
  decode.Work_mode="3DS18B20";
  decode.TempC2=parseFloat(((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
  decode.TempC3=parseFloat(((bytes[9]<<8 | bytes[10])/10) .toFixed(1));

}
else if(mode=='4')
{
  decode.Work_mode="Weight";
  decode.Weight=(bytes[7]<<24>>16 | bytes[8]);
}
else if(mode=='5')
{
  decode.Work_mode="Count";
  decode.Count=(bytes[7]<<24 | bytes[8]<<16 | bytes[9]<<8 | bytes[10]);
```

```
    decode.Count=(bytes[7]<<24 | bytes[8]<<16 | bytes[9]<<8 | bytes[10]);
}

  if((bytes.length==11)||(bytes.length==12))
  {
  msg.payload=decode;
  return msg;
  //return decode;

  }
//}
```

```
var bytes =msg.payload;


//function Decoder(bytes, port) {
var mode=(bytes[6] & 0x7C)>>2;
var decode = {};
if(mode!=2)
{
 decode.BatV=(bytes[0]<<8 | bytes[1])/1000;
 decode.TempC1= parseFloat(((bytes[2]<<24>>16 | bytes[3])/10).toFixed(2));
 decode.ADC_CH0V=(bytes[4]<<8 | bytes[5])/1000;
 decode.Digital_IStatus=(bytes[6] & 0x02)? "H":"L";
 if(mode!=6)
 {
          decode.EXTI_Trigger=(bytes[6] & 0x01)? "TRUE":"FALSE";
 decode.Door_status=(bytes[6] & 0x80)? "CLOSE":"OPEN";
 }
}


if(mode=='0')
{
 decode.Work_mode="IIC";
 if((bytes[9]<<8 | bytes[10])===0)
 {
  decode.Illum=(bytes[7]<<24>>16 | bytes[8]);
 }
 else
```

```
  {
  decode.TempC_SHT=parseFloat(((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
  decode.Hum_SHT=parseFloat(((bytes[9]<<8 | bytes[10])/10).toFixed(1));
  }
}
else if(mode=='1')
{
  decode.Work_mode=" Distance";
  decode.Distance_cm=parseFloat(((bytes[7]<<8 | bytes[8])/10) .toFixed(1));
  if((bytes[9]<<8 | bytes[10])!=65535)
  {
    decode.Distance_signal_strength=parseFloat((bytes[9]<<8 | bytes[10]) .toFixed(0));
  }
}
else if(mode=='2')
{
  decode.Work_mode=" 3ADC";
  decode.BatV=bytes[11]/10;
  decode.ADC_CH0V=(bytes[0]<<8 | bytes[1])/1000;
  decode.ADC_CH1V=(bytes[2]<<8 | bytes[3])/1000;
  decode.ADC_CH4V=(bytes[4]<<8 | bytes[5])/1000;
  decode.Digital_IStatus=(bytes[6] & 0x02)? "H":"L";
  decode.EXTI_Trigger=(bytes[6] & 0x01)? "TRUE":"FALSE";
  decode.Door_status=(bytes[6] & 0x80)? "CLOSE":"OPEN";
  if((bytes[9]<<8 | bytes[10])===0)
  {
    decode.Illum=(bytes[7]<<24>>16 | bytes[8]);
  }
  else
  {
  decode.TempC_SHT=parseFloat(((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
```

```
  decode.Hum_SHT=parseFloat(((bytes[9]<<8 | bytes[10])/10) .toFixed(1));
  }
}
else if(mode=='3')
{
  decode.Work_mode="3DS18B20";
  decode.TempC2=parseFloat(((bytes[7]<<24>>16 | bytes[8])/10).toFixed(2));
  decode.TempC3=parseFloat(((bytes[9]<<8 | bytes[10])/10) .toFixed(1));

}
else if(mode=='4')
{
  decode.Work_mode="Weight";
  decode.Weight=(bytes[7]<<24>>16 | bytes[8]);
}
else if(mode=='5')
{
  decode.Work_mode="Count";
  decode.Count=(bytes[7]<<24 | bytes[8]<<16 | bytes[9]<<8 | bytes[10]);
}

  if((bytes.length==11)||(bytes.length==12))
  {
  msg.payload=decode;
  return msg;
  //return decode;

  }
//}
```