

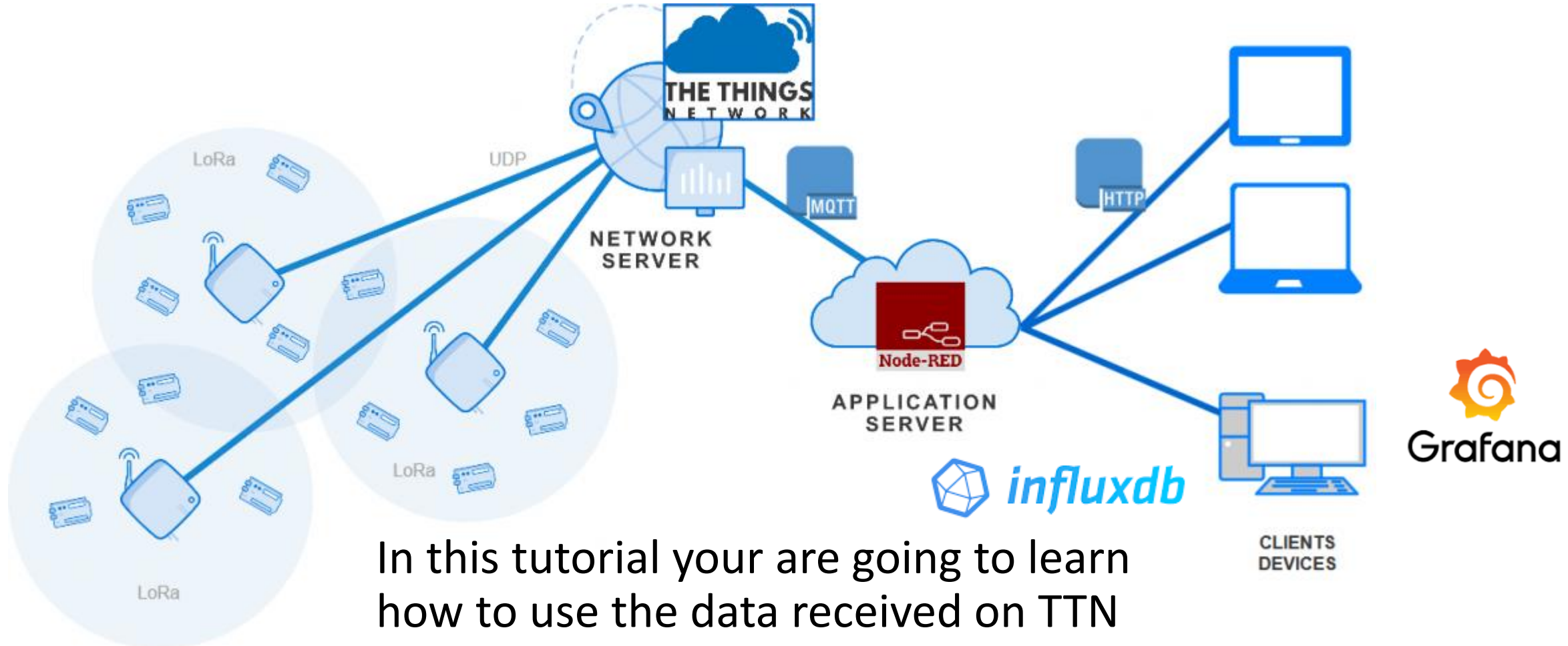
IoT LoRa application service Tutorial

F. Ferrero, J. Lanteri

V.1.3



Node Red – InFluxDB - GRAFANA

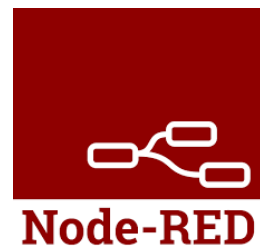


Outline

1/ Definition

2/ Tutorial

Node Red



- Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.
- It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.
- Built on Node.js
 - The light-weight runtime is built on Node.js, taking full advantage of its event-driven, non-blocking model. This makes it ideal to run at the edge of the network on low-cost hardware such as the Raspberry Pi as well as in the cloud.

Outline

1/ Definition

2/ Tutorial

Node-Red

First install Node.JS : <https://nodejs.org/en/download/>

Then : Install Node-red :

<https://nodered.org/docs/getting-started/windows>

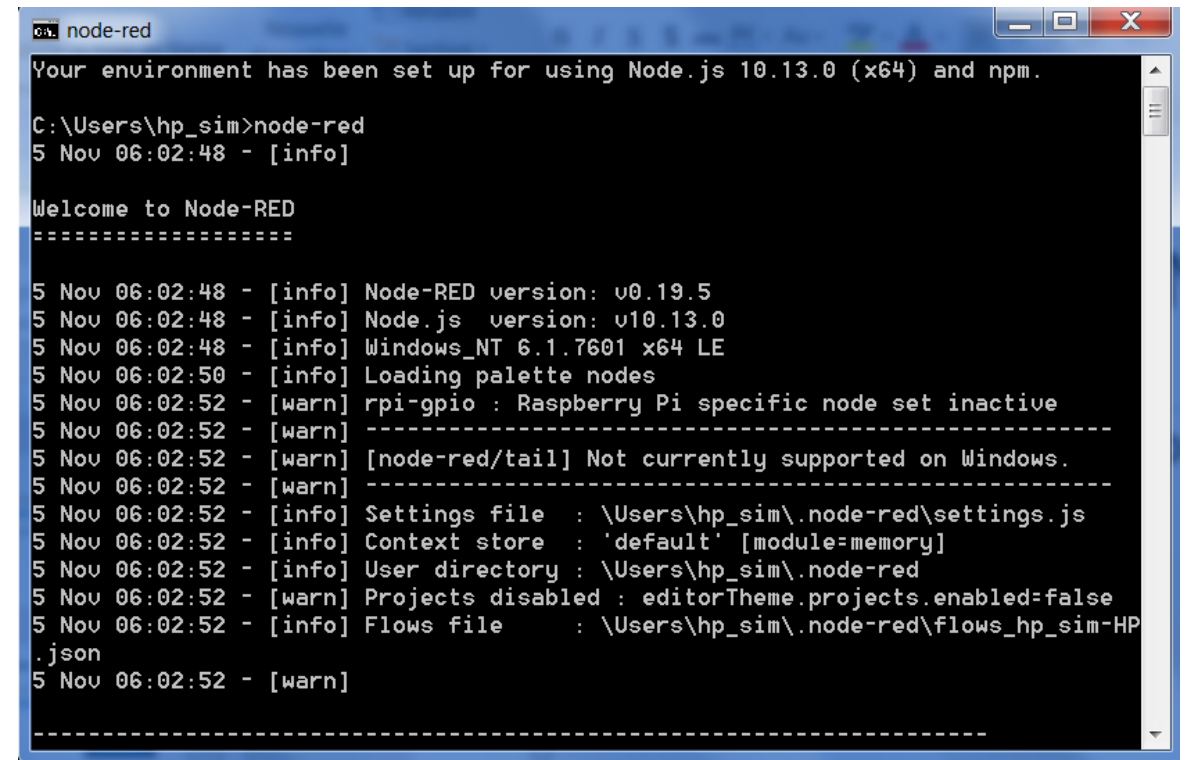
Install package in Node Red :

node-red-node-email

Install git : <https://git-scm.com/downloads>

Connecting to TTN

- Start NODE.js command prompt
- Run : node-red
- Open your web browser and go to <http://127.0.0.1:1880>



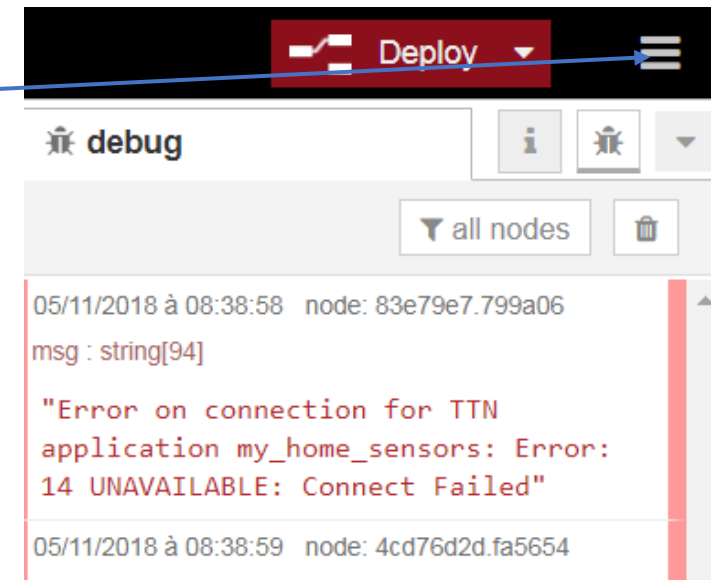
```
node-red
Your environment has been set up for using Node.js 10.13.0 (x64) and npm.

C:\Users\hp_sim>node-red
5 Nov 06:02:48 - [info]

Welcome to Node-RED
=====

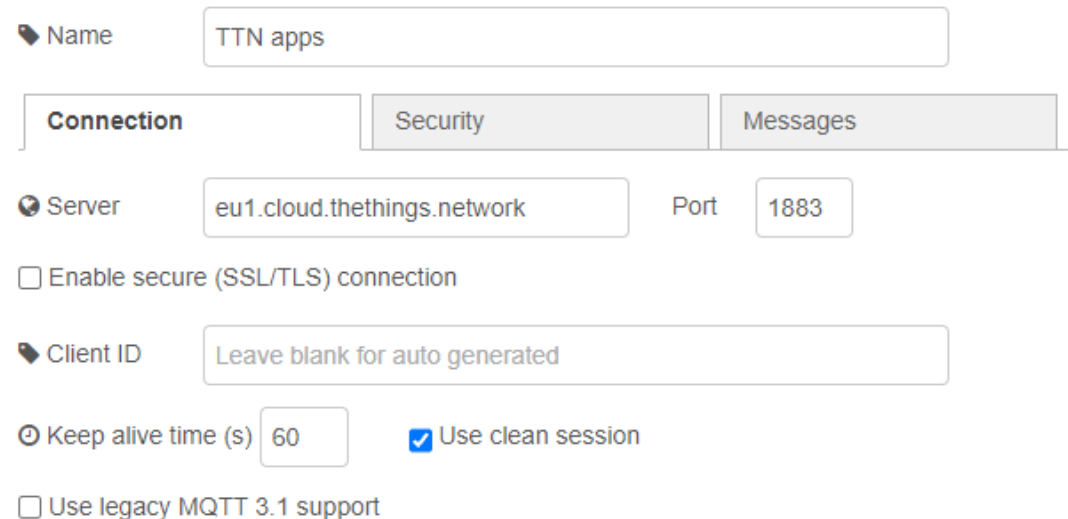
5 Nov 06:02:48 - [info] Node-RED version: v0.19.5
5 Nov 06:02:48 - [info] Node.js version: v10.13.0
5 Nov 06:02:48 - [info] Windows_NT 6.1.7601 x64 LE
5 Nov 06:02:50 - [info] Loading palette nodes
5 Nov 06:02:52 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
5 Nov 06:02:52 - [warn] -----
5 Nov 06:02:52 - [warn] [node-red/tail] Not currently supported on Windows.
5 Nov 06:02:52 - [warn] -----
5 Nov 06:02:52 - [info] Settings file : \Users\hp_sim\.node-red\settings.js
5 Nov 06:02:52 - [info] Context store : 'default' [module=memory]
5 Nov 06:02:52 - [info] User directory : \Users\hp_sim\.node-red
5 Nov 06:02:52 - [warn] Projects disabled : editorTheme.projects.enabled=false
5 Nov 06:02:52 - [info] Flows file : \Users\hp_sim\.node-red\flows_hp_sim-HP
.json
5 Nov 06:02:52 - [warn] -----
```

- On the editor, click here
- And go to palette editor
- Install :
- node-red-contrib-ttn



Connecting to TTN

- You have the graphical Node-red editor
- Add mqtt in node
- Edit mqtt
- Choose « Add new mqtt-broker ... » in App and click on edit
- Output is a parsed JSON object



TTN apps

Connection | Security | Messages

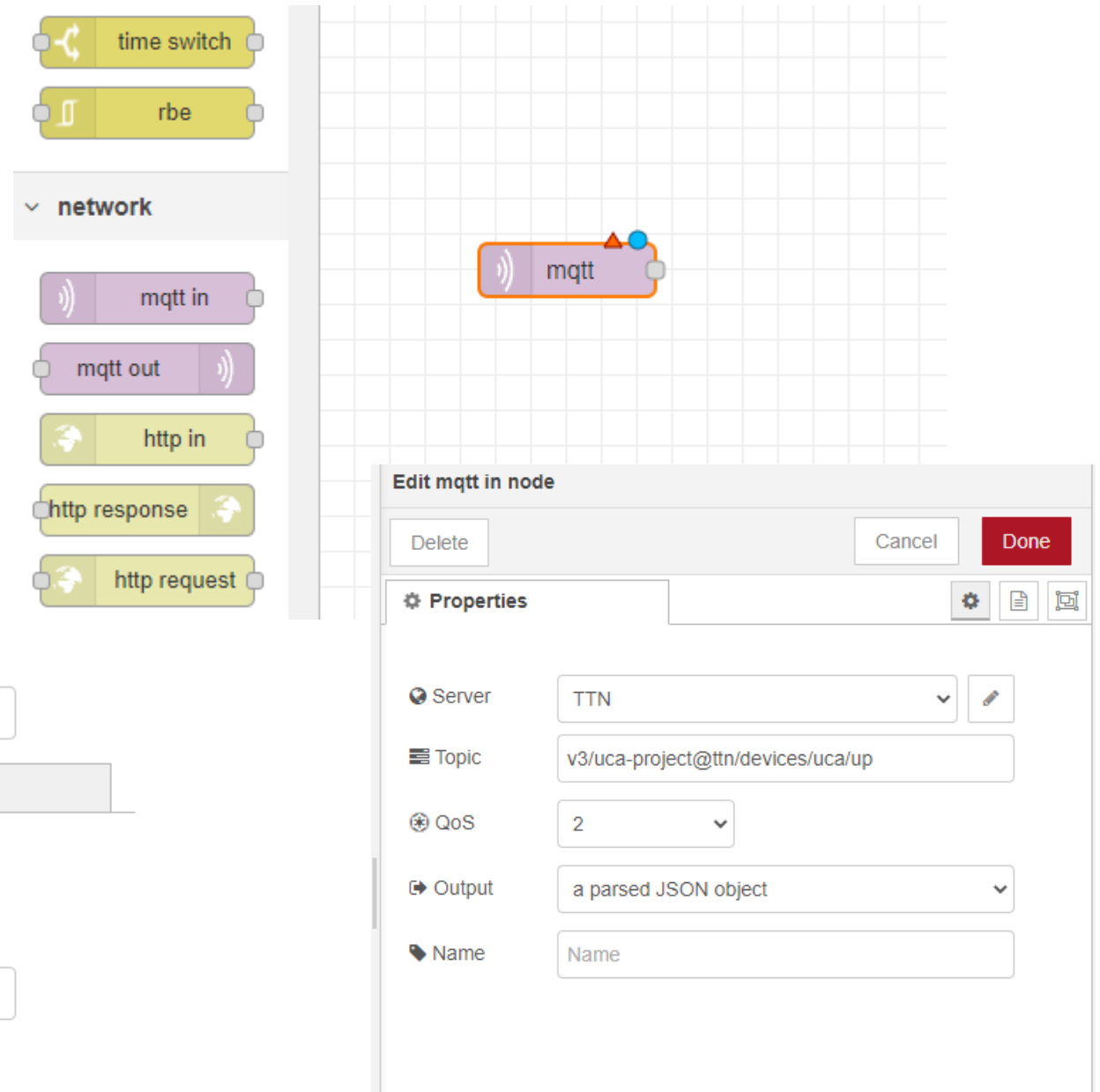
Server: eu1.cloud.thethings.network Port: 1883

☐ Enable secure (SSL/TLS) connection

Client ID: Leave blank for auto generated

Keep alive time (s): 60 ☒ Use clean session

☐ Use legacy MQTT 3.1 support



time switch

rbe

network

mqtt in

mqtt out

http in

http response

http request

mqtt

Edit mqtt in node

Delete Cancel Done

Properties

Server: TTN

Topic: v3/uca-project@ttn/devices/uca/up

QoS: 2

Output: a parsed JSON object

Name: Name

Connecting to TTN

- Update security and topic:
discovery.thethingsnetwork.org:1900

- Go to you application in TTN
- Copy past the User name and keys

v3/uca-project@ttn/devices/**device_name**/up

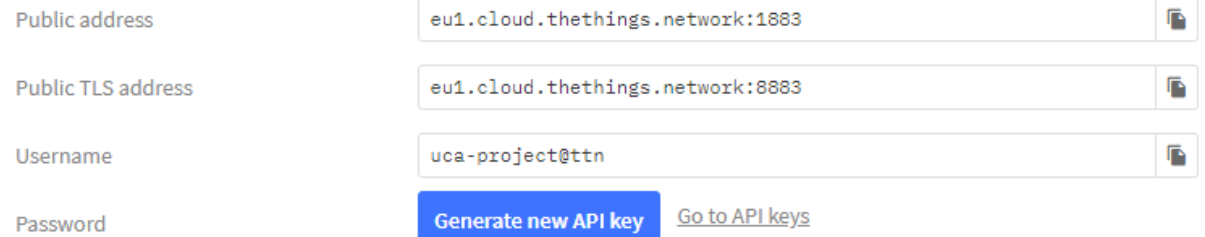


The screenshot shows the TTN application configuration interface. At the top, there is a 'Name' field with a small icon. Below it, there are three tabs: 'Connection', 'Security' (which is selected and highlighted), and 'Messages'. Under the 'Security' tab, there are two input fields: 'Username' and 'Password'. The 'Password' field is masked with dots.

MQTT

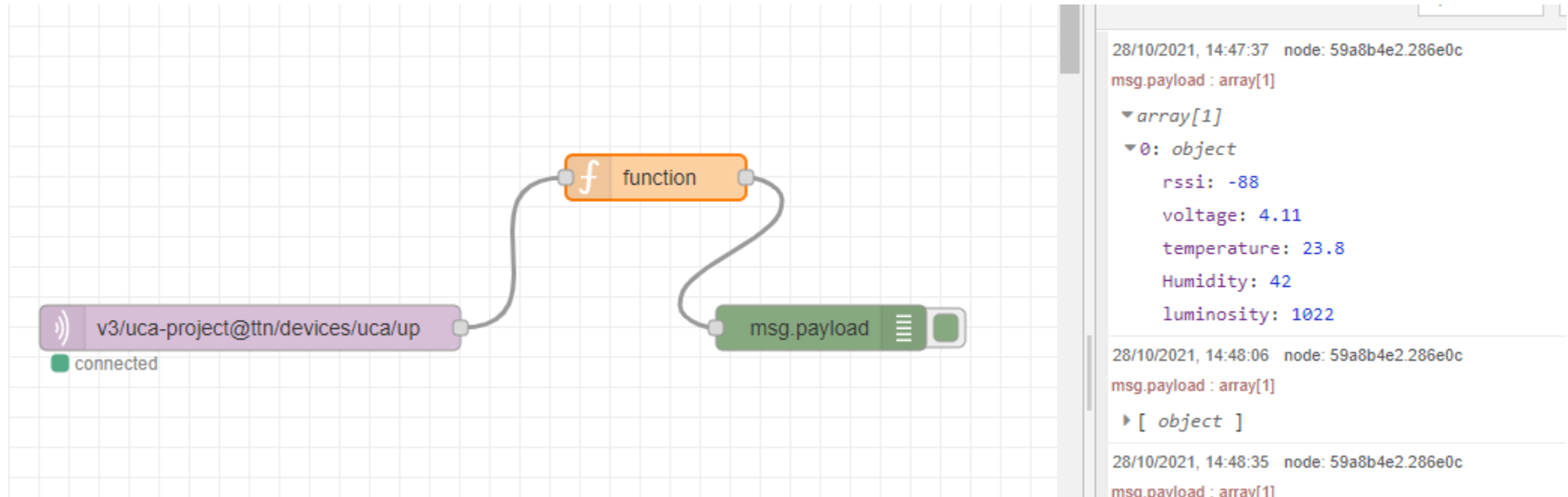
The Application Server exposes an MQTT server to work with streaming events. In order to use the MQTT server you need to create a new API key, which will function as connection password. You can also use an existing API key, as long as it has the necessary rights granted. Use the connection information below to connect.

Connection credentials



The screenshot shows the 'Connection credentials' section of the TTN interface. It contains four input fields, each with a copy icon on the right: 'Public address' (eu1.cloud.thethings.network:1883), 'Public TLS address' (eu1.cloud.thethings.network:8883), 'Username' (uca-project@ttn), and 'Password'. The 'Password' field is highlighted with a blue button labeled 'Generate new API key' and a link labeled 'Go to API keys'.

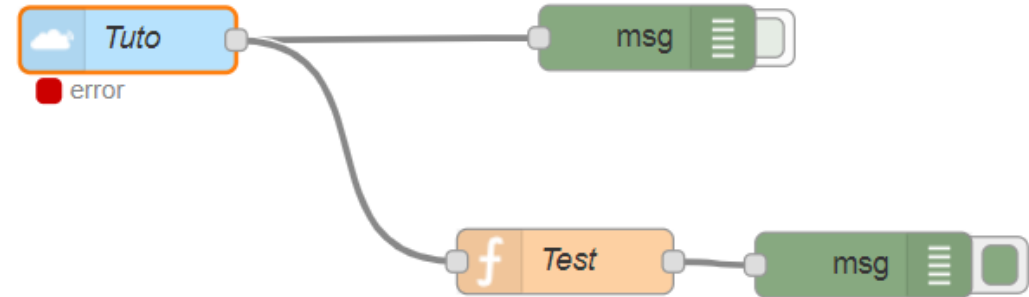
Connecting to TTN



```
var array = [];
array = [{
  rssi: msg.payload.uplink_message.rx_metadata[0].rssi, voltage:
  msg.payload.uplink_message.decoded_payload.analog_in_3,
  temperature:msg.payload.uplink_message.decoded_payload.temperature_1, Humidity:
  msg.payload.uplink_message.decoded_payload.relative_humidity_2, luminosity:
  msg.payload.uplink_message.decoded_payload.luminosity_4, }];var msg1 =
{};msg1.payload = array;return [msg1];
```

Connecting to TTN

- If you want to extract only 1 data,
- As an exemple the RSSI (received signal Strength indicator
- Use a function to extract the desired data



```
var tmp = {};  
tmp.payload = msg.payload.luminosity_4;  
return tmp;
```

14/03/2021, 21:36:02 node: e0e31eb9.79c5e

msg : Object

▼ object

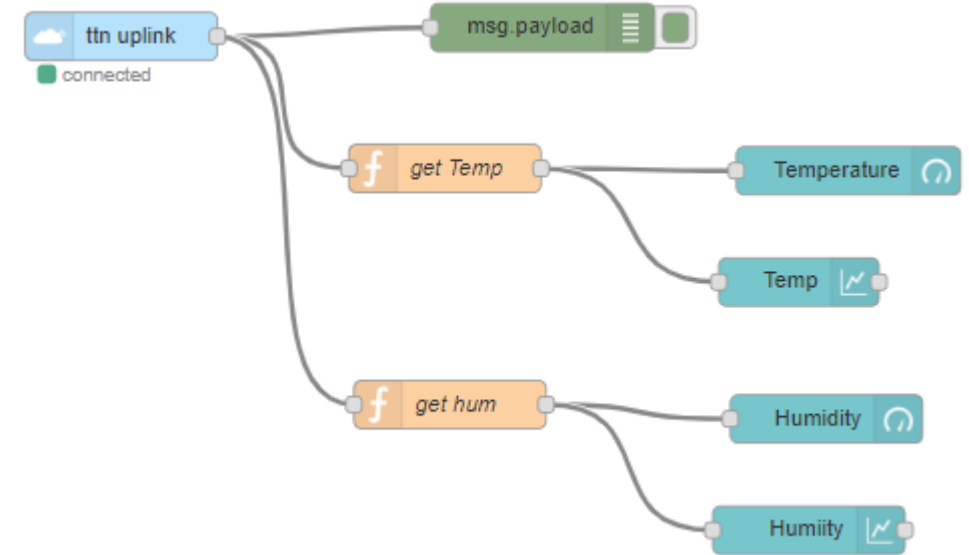
payload: 241

_msgid: "572f7153.ffebf"

Add a Dashboard

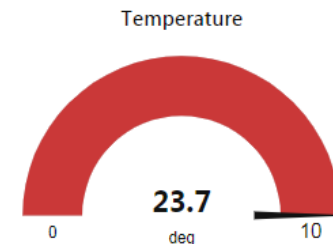
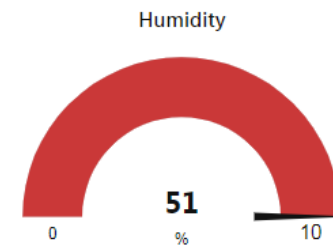
- Go to Manage Palette, select Install
- Install : node-red-dashboard
- Add a function to extract sensor values (Temp, Hum, luminosity...)
- Add Gauge and Graph for Dashboard section
- Add a new UI group in the Gauge and Graph
- Go to : <http://127.0.0.1:1880/ui/>

```
var tmp = {};  
tmp.payload = msg.payload.luminosity_4;  
return tmp;
```

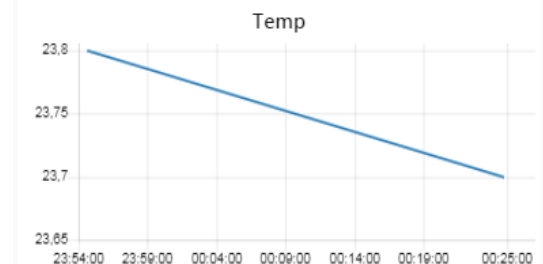
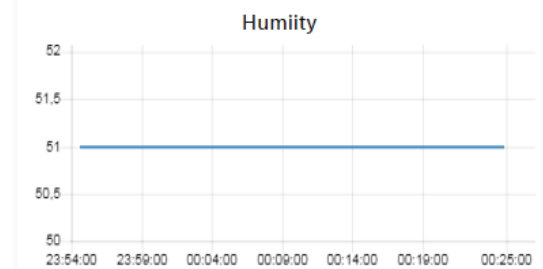


```
1 var tmp = {};  
2 tmp.payload = msg.payload.luminosity_4;  
3 return tmp;  
4
```

test

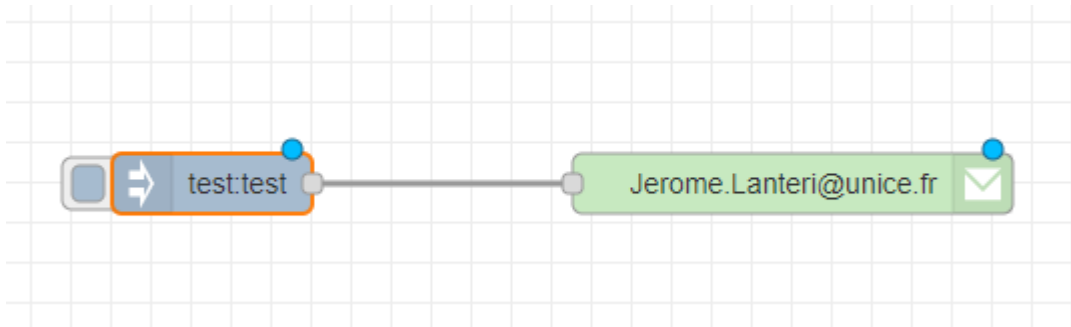


Test



Send an email

- Go to Manage Palette, select Install
- Install : node-red-node-email
- Configure your email with unice credential (use your ENT account)
- Use a timestamp to test (click to trigger it)
- Email object is defined in msg.topic
- Email content is defined in msg.payload



Edit email node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖼

✉ To

Jerome.Lanteri@unice.fr

🌐 Server

smtp.unice.fr

🔌 Port

587

☐ Use secure connection.

👤 Userid

ferrero@unice.fr

🔒 Password

.....

🔒 Use TLS?

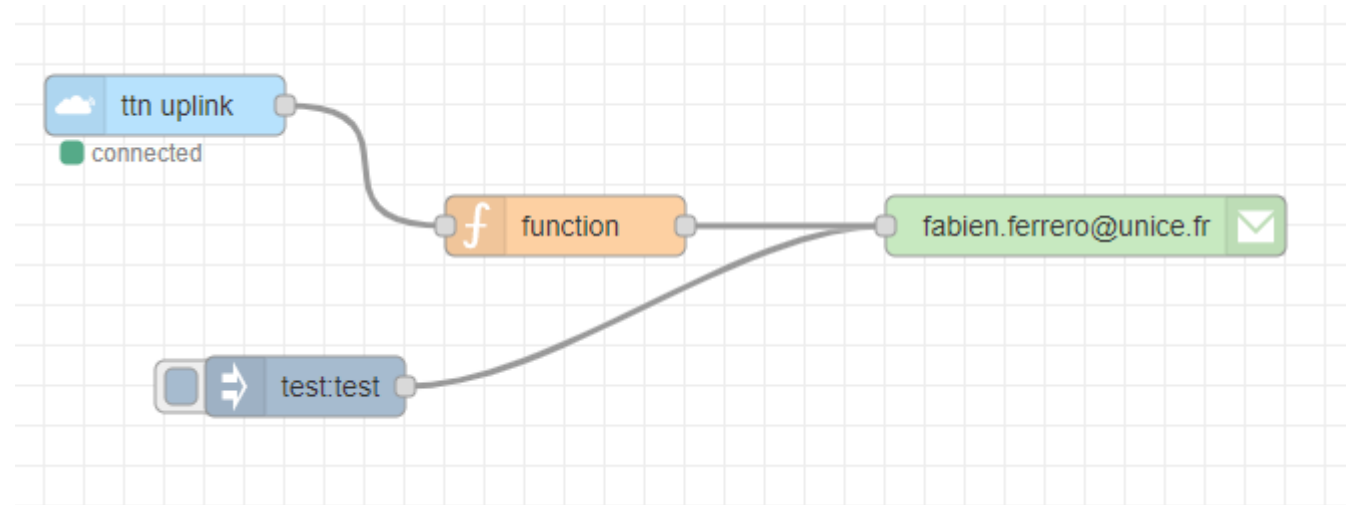
☒

🏷 Name

Name

Send an email triggered on luminosity level

- Add a function to detect if the luminosity overpass a threshold
- Send an email with a sentence



```
1 var tmp = {};  
2 var lum = msg.payload.luminosity_4;  
3 tmp.topic = "Information sur le capteur";  
4 if (lum > 100)  
5   tmp.payload = "il y a de la lumière";  
6 else  
7   tmp.payload = "il n'y a pas de lumière";  
8 return tmp;  
9  
10
```

Principale			Réseaux sociaux	Promotions
<input type="checkbox"/>	☆	»	Fabien.Ferrero	Information sur le capteur - il n'y a pas de lumière
<input type="checkbox"/>	☆	»	Fabien.Ferrero	Information sur le capteur - il n'y a pas de lumière
<input type="checkbox"/>	☆	»	Fabien.Ferrero	Information sur le capteur - il y a de la lumière
<input type="checkbox"/>	☆	»	Fabien.Ferrero	Information sur le capteur - il y a de la lumière
<input type="checkbox"/>	☆	»	Fabien.Ferrero	Information sur le capteur - il y a de la lumière

Good luck for you projects !

