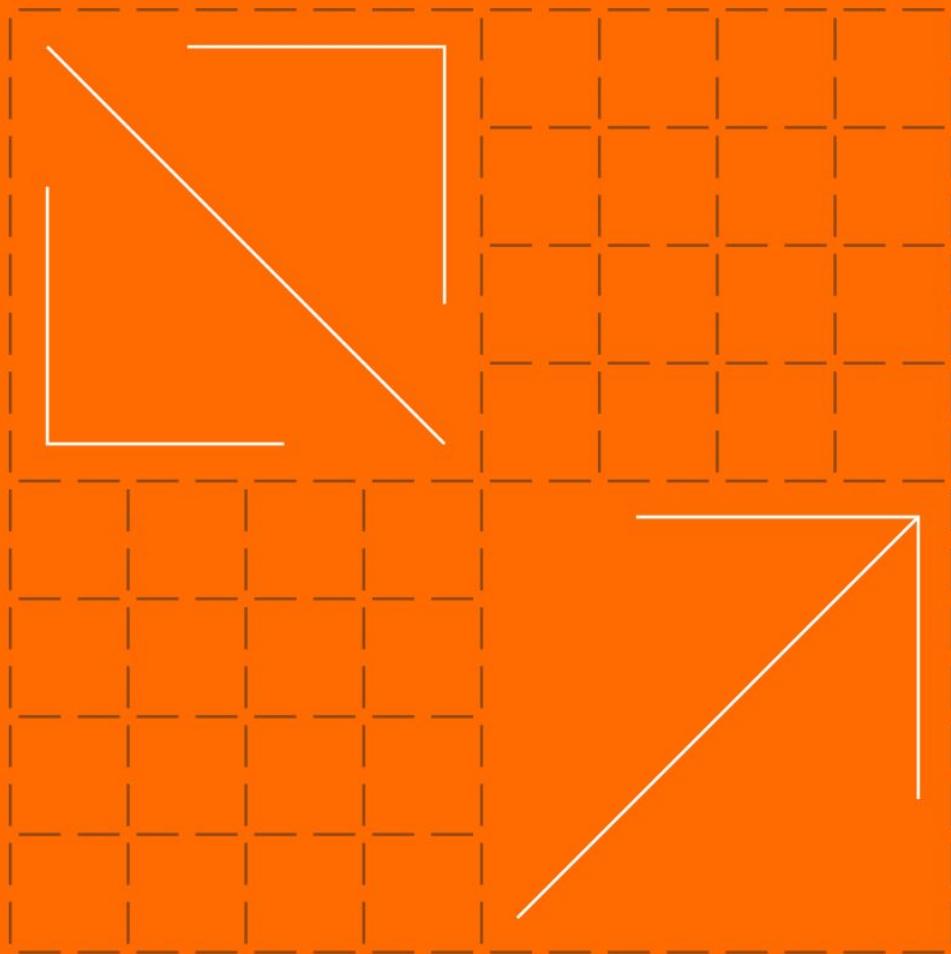


Learn what is LoRa, LoRaWAN and build your first LoRaWAN gateway, the easy way™

Marc Pous - Developer Advocate at balena.io

8th November 2022

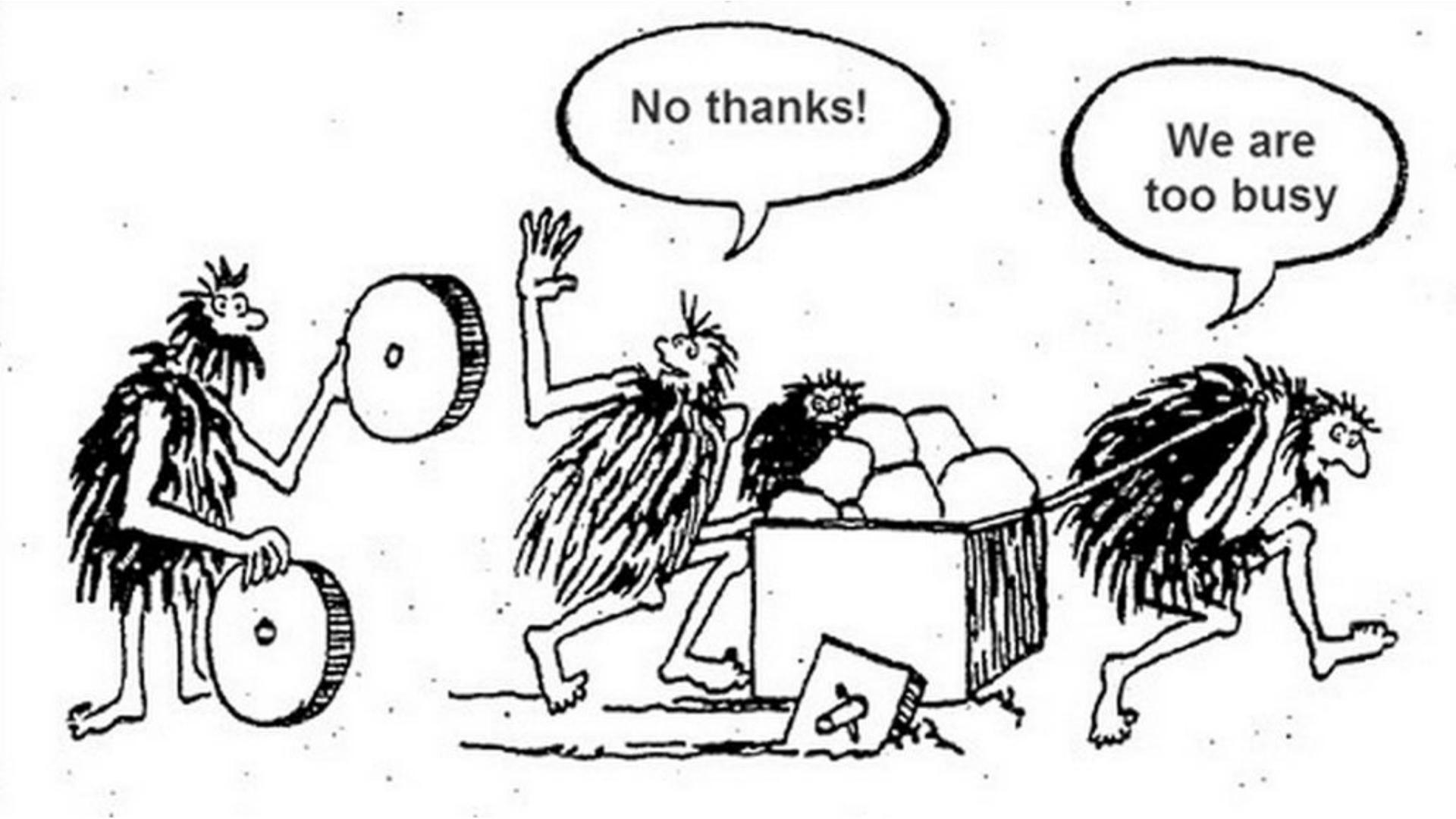




Marc Pous

Developer Advocate @ **balena.io**
Initiator **IoT Barcelona** and **IoT Stars**

e: marc@balena.io
t: [@gy4nt](https://twitter.com/@gy4nt)
g: github.com/mpous



No thanks!

We are
too busy

LPWAN

Low Power Wide Area Networks

Canada

END

UNITED
STATES

Mexico

Cuba



seeed studio



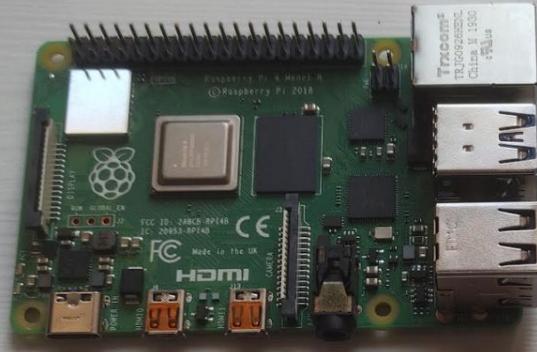
S2101

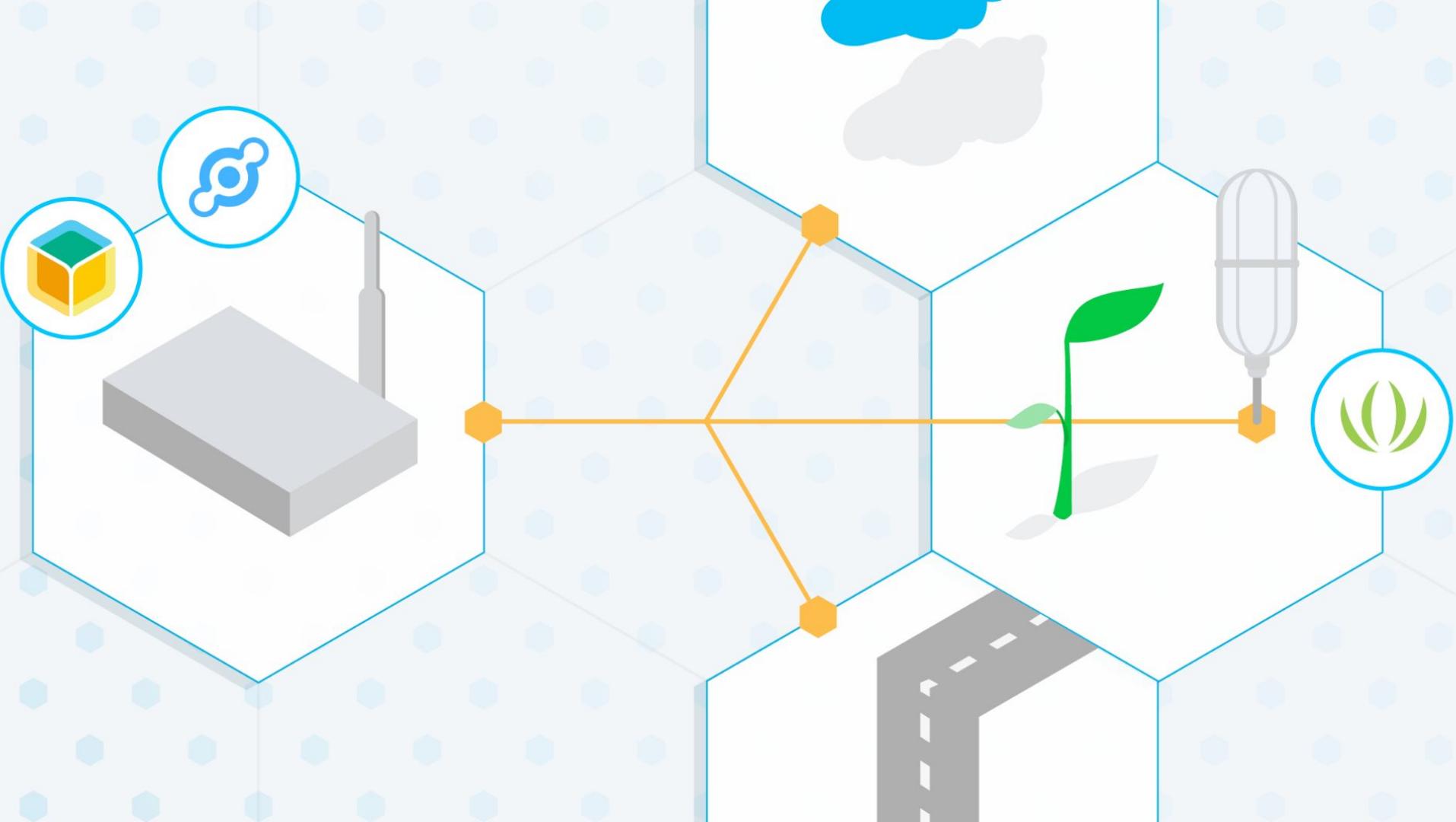


S2104

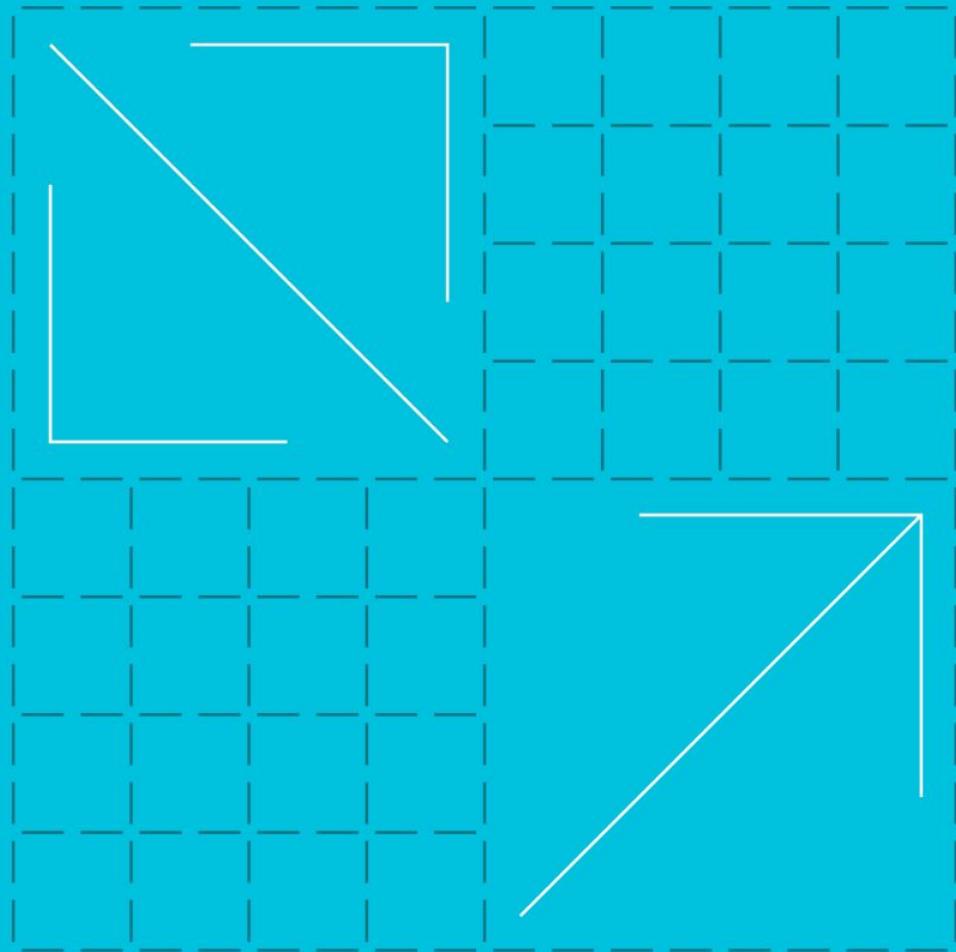
<https://www.seeedstudio.com/SenseCAP-S2101-LoRaWAN-Air-Temperature-and-Humidity-Sensor-p-5354.html>

<https://www.seeedstudio.com/SenseCAP-S2104-LoRaWAN-Soil-Temperature-and-Moisture-Sensor-p-5357.html>





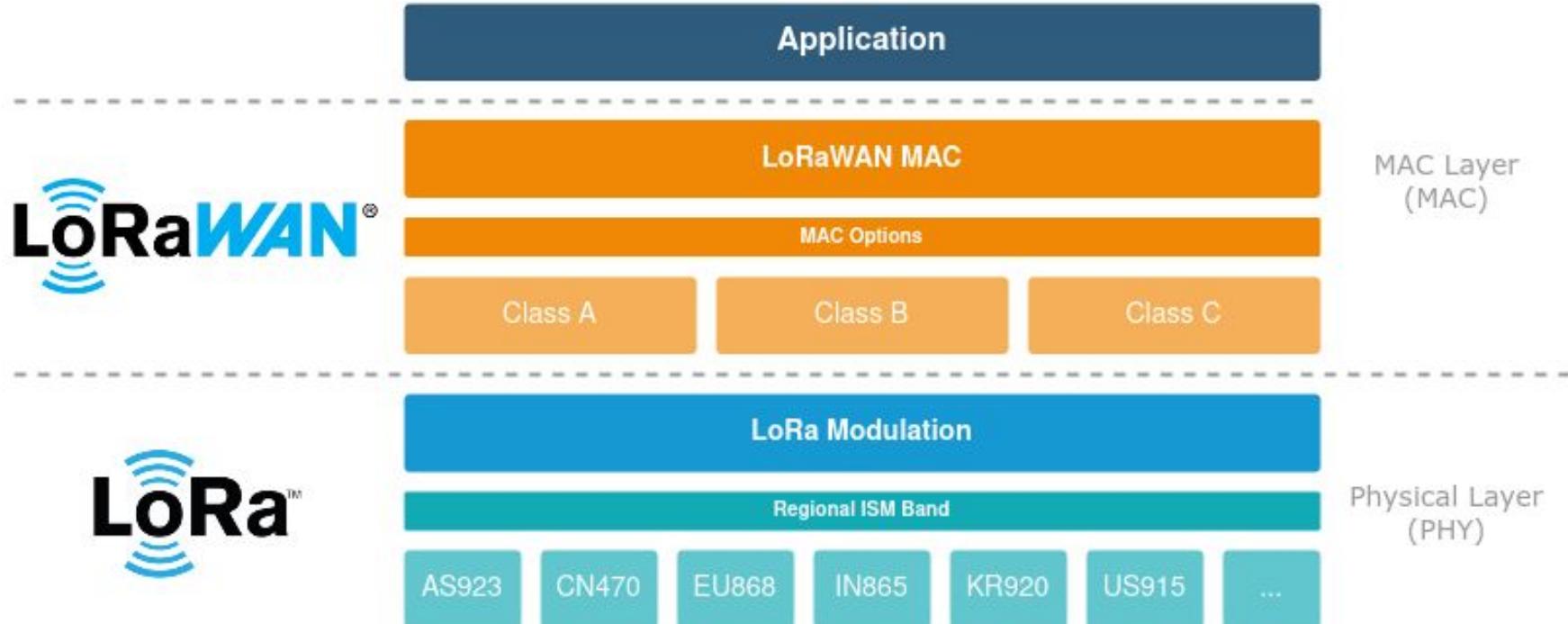
LoRa and LoRaWAN?

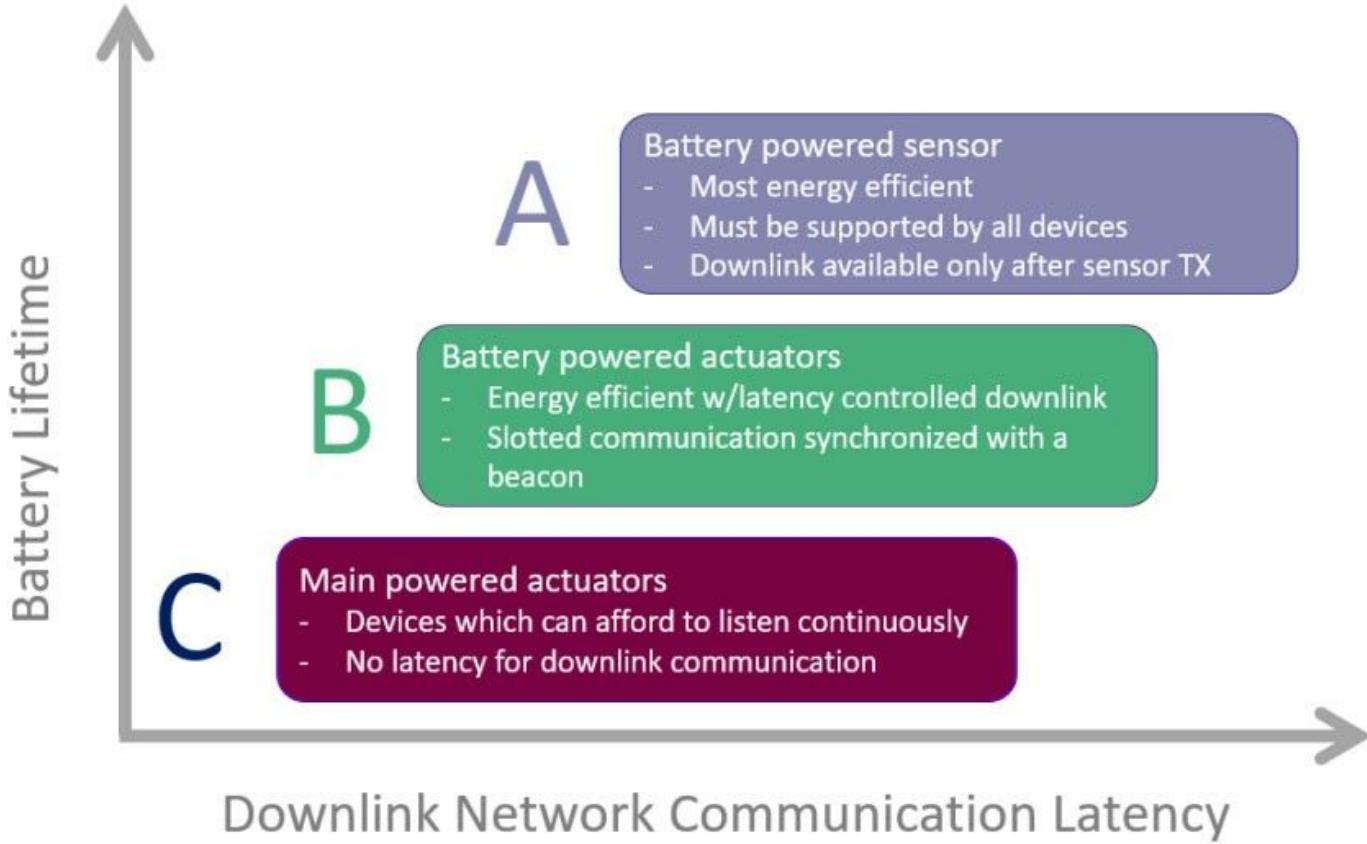


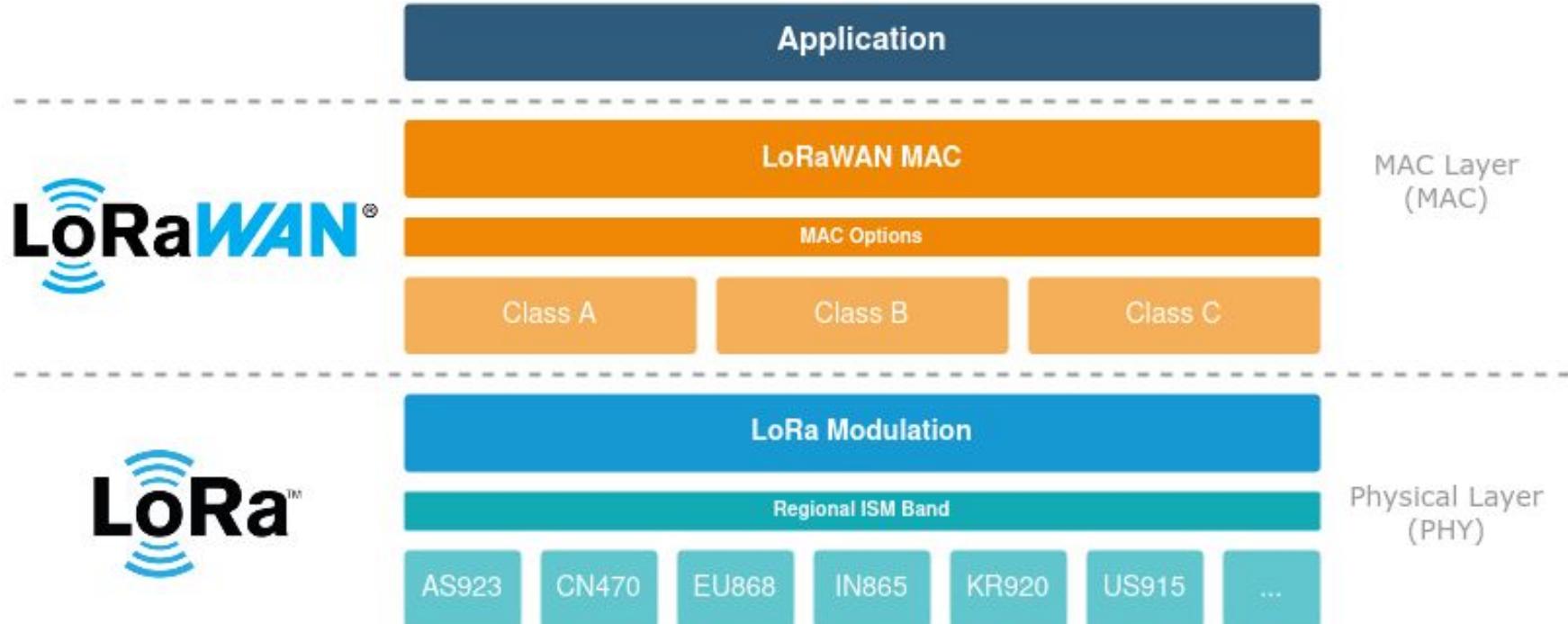
LoRa is a radio modulation meant for Long Range communication using very little power.

LoRa is a radio modulation meant for Long Range communication using very little power.

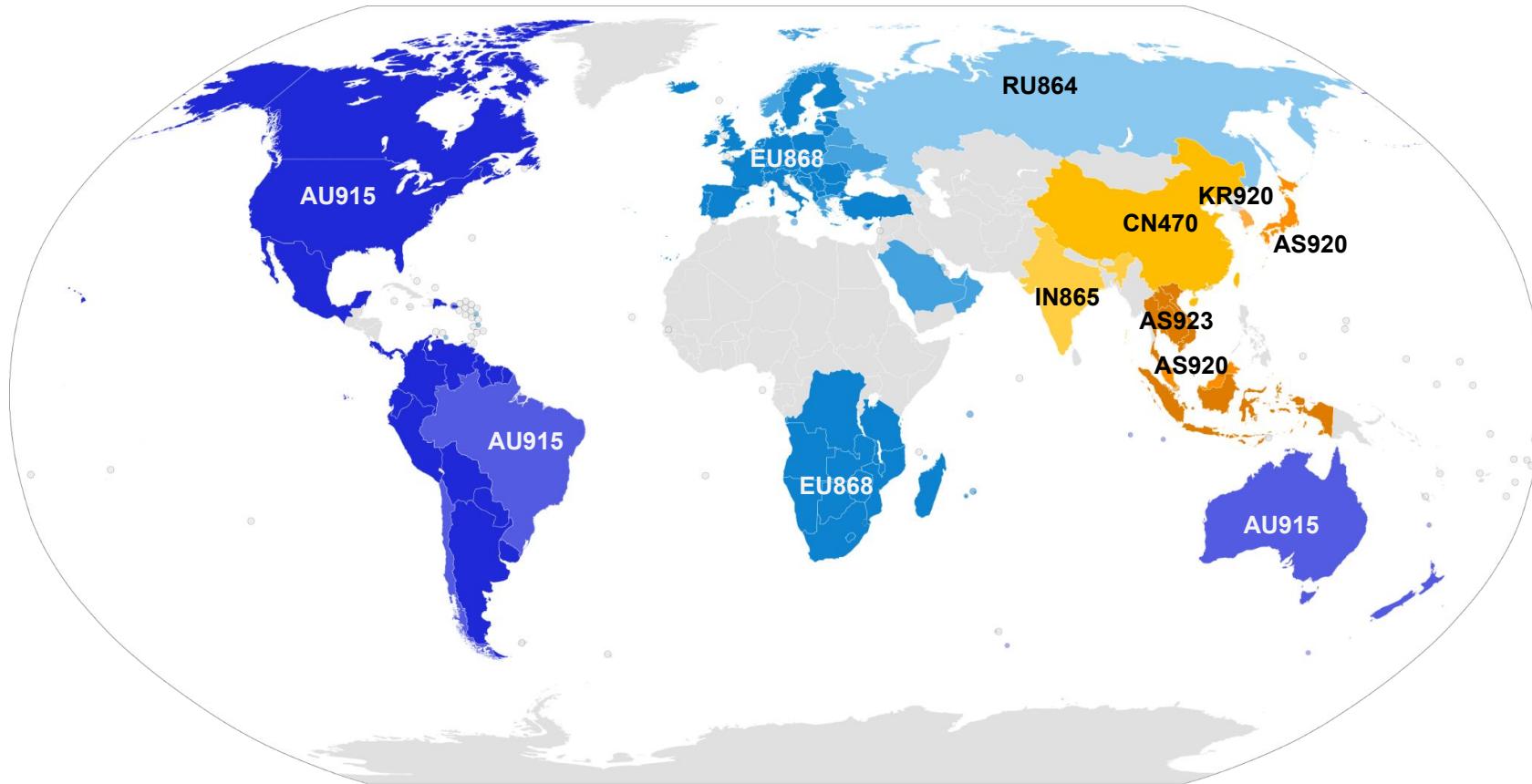
LoRaWAN is a Media Access Layer (MAC) protocol. A **software layer** that defines how devices use LoRa to transmit and receive messages on a network.



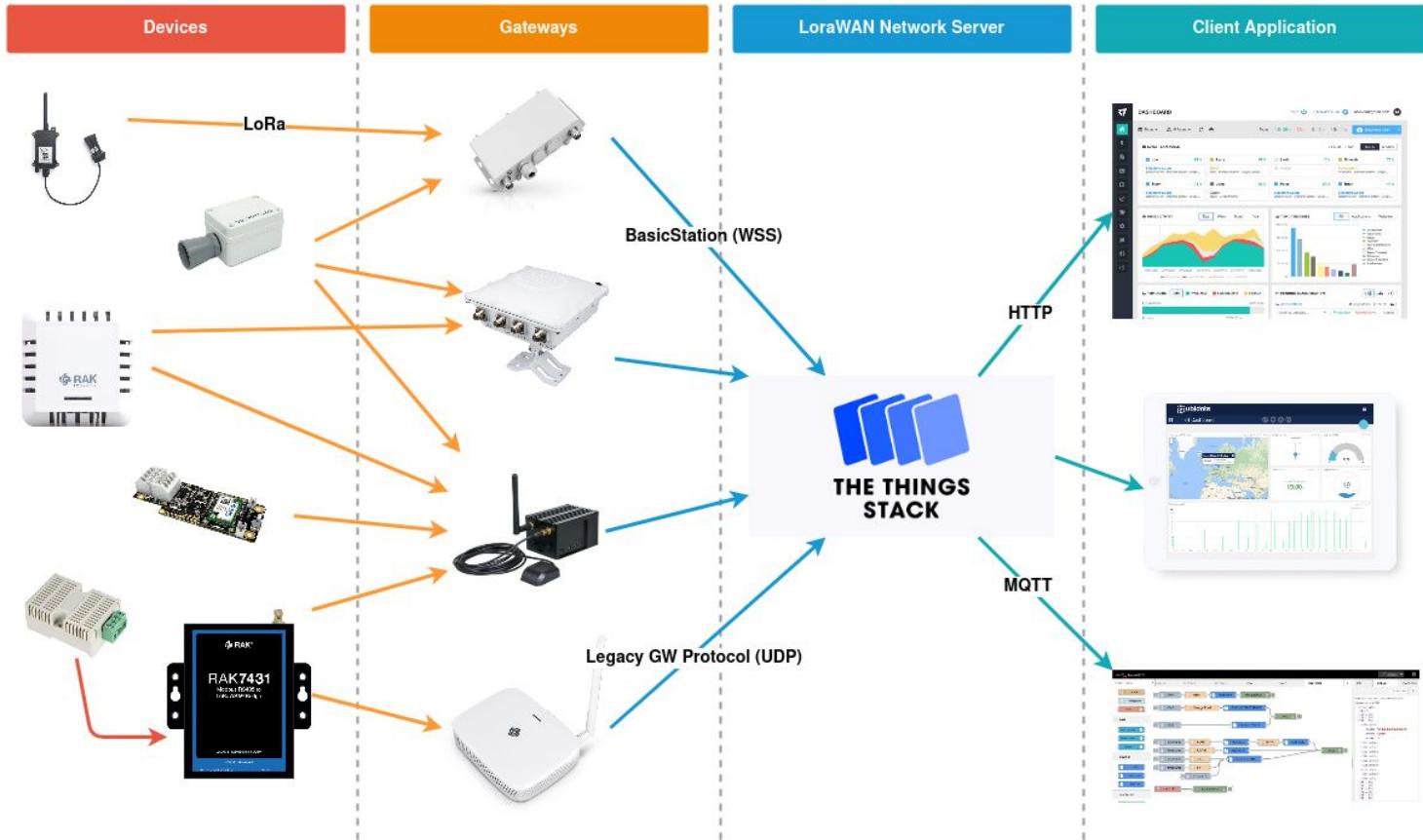




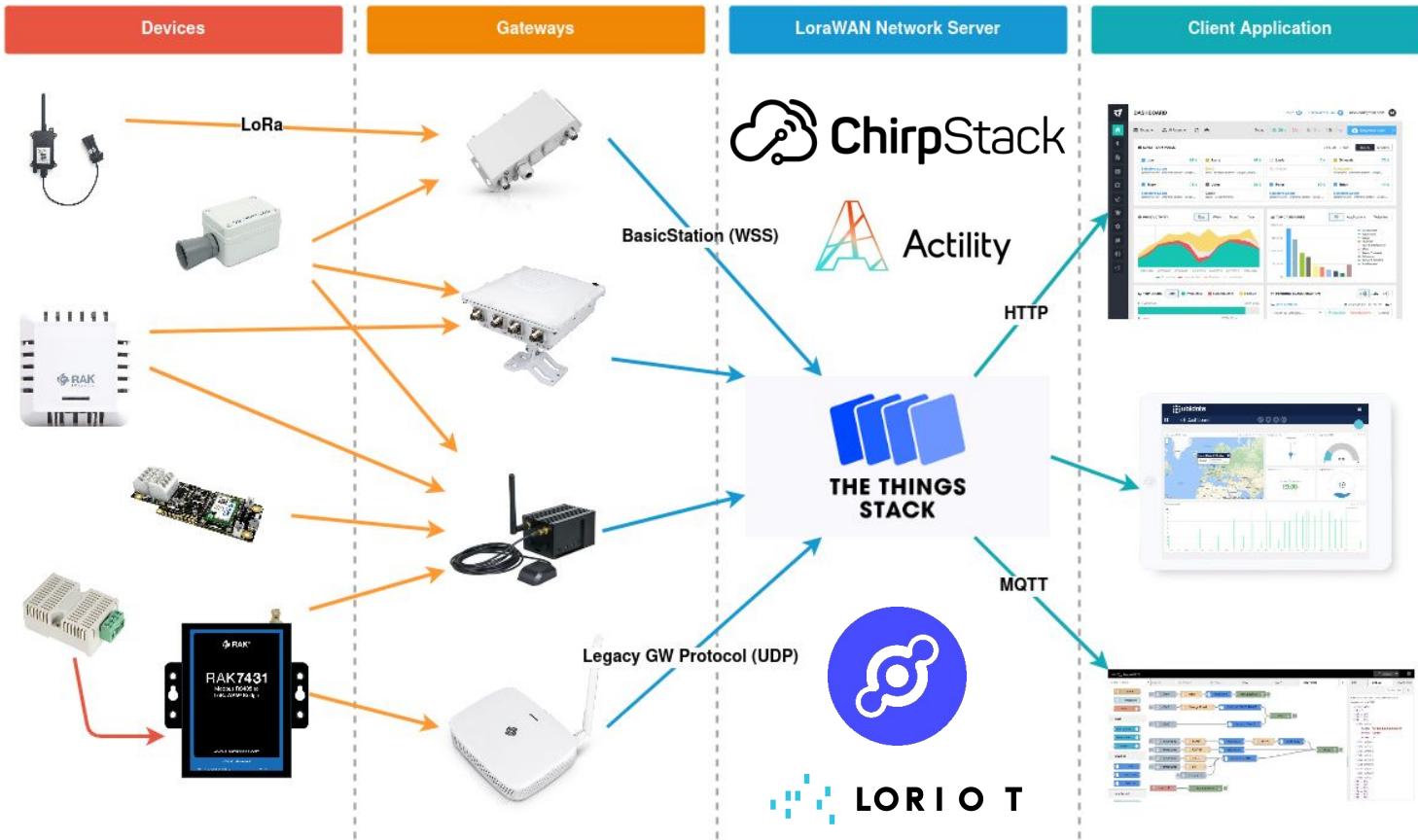
Regional frequencies



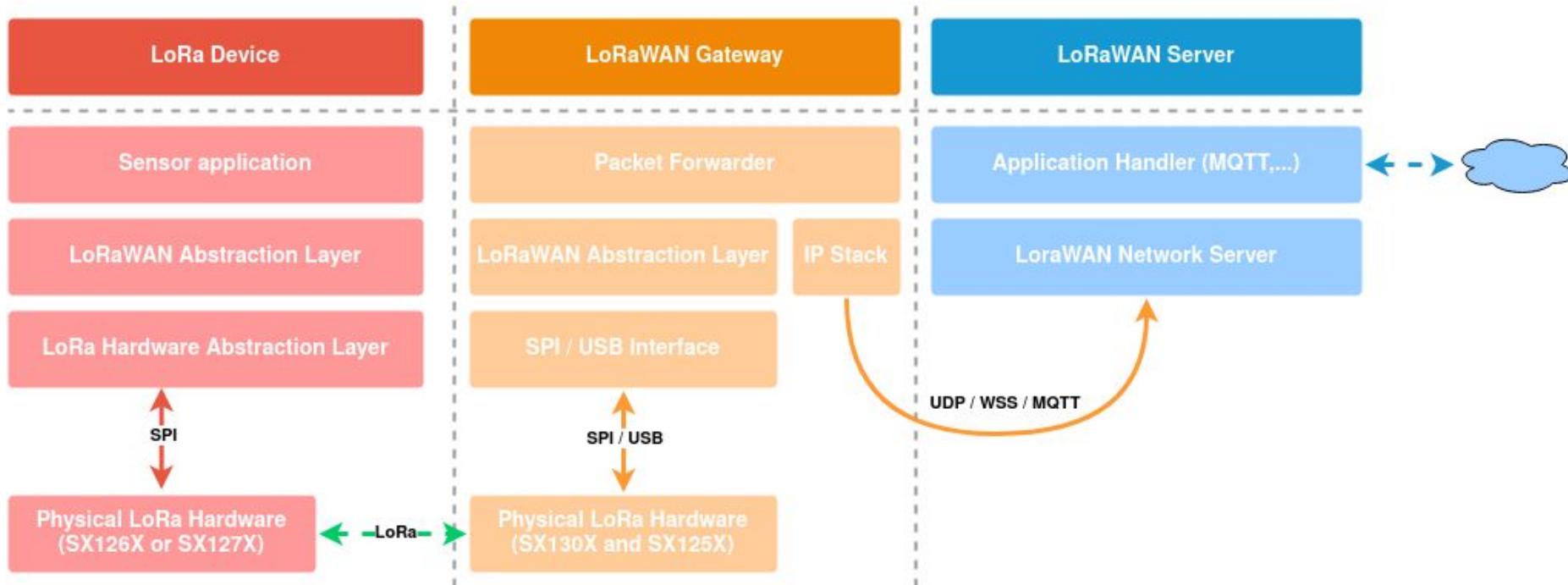
LoRaWAN Network Architecture



LoRaWAN Network Architecture



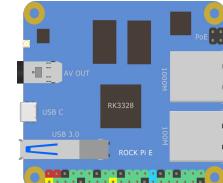
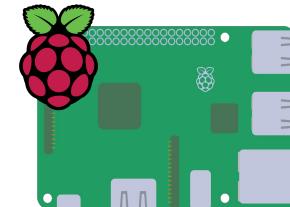
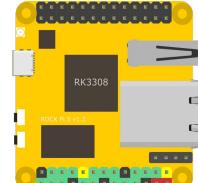
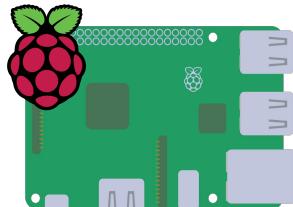
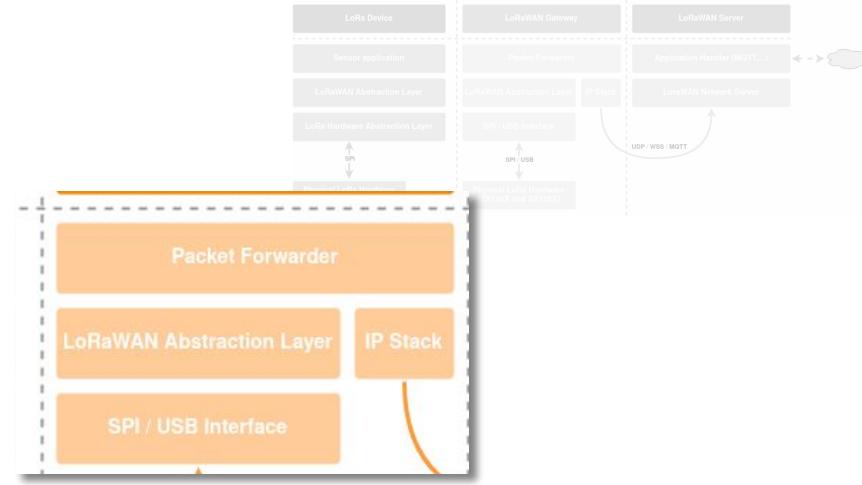
LoRaWAN Network Architecture



The Host

The **host** is usually a Linux machine responsible to run the LoRa and LoRaWAN abstraction layers and the packet forwarder service to forward packets to the LoRaWAN Network Server (**LNS**).

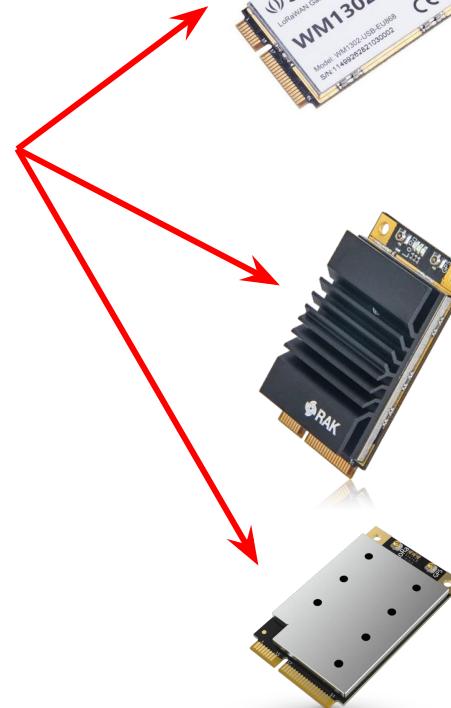
In this workshop we will use a **Raspberry Pi** running **balenaOS** with a **Basics™ Station** service as packet forwarder (and more containers).



LoRa concentrator



RAK2245
(SX1301)



WM1302
(SX1302)



RAK2287
(SX1302)



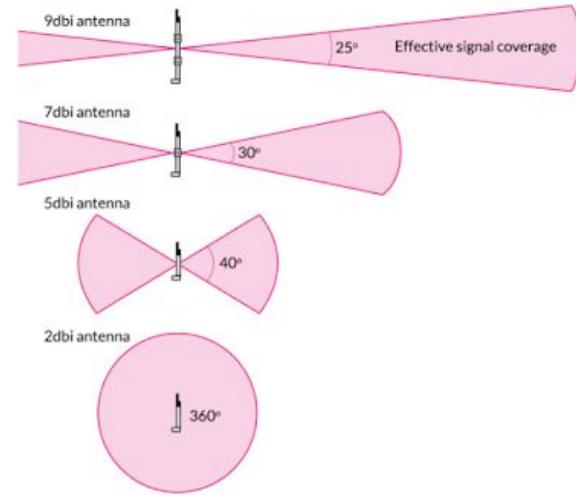
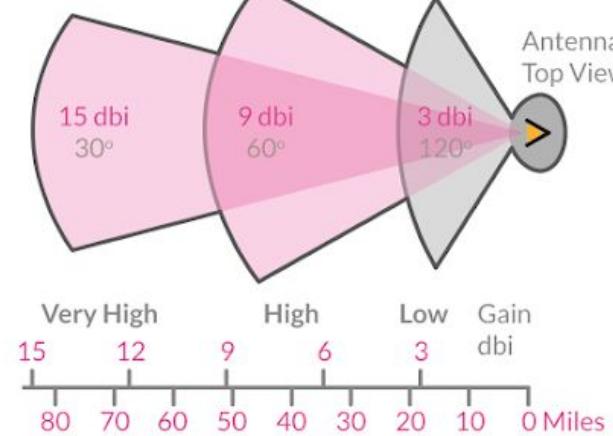
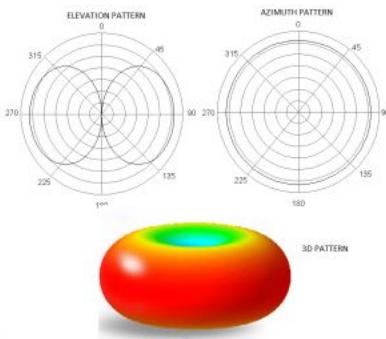
RAK5146
(SX1303)



The Antenna - Gain and directionality

An antenna does not “create” energy, it just **radiates it unevenly depending on the direction**. The gain is the relative radiated power in the direction of maximum radiated power related to an isotropic antenna. The more gain, the more directionality.

Prior to choose the gain we want, we should analyze the node distribution around the antenna.



The Antenna - Connectors



TNC



BNC



Type N



SMA



RP-SMA



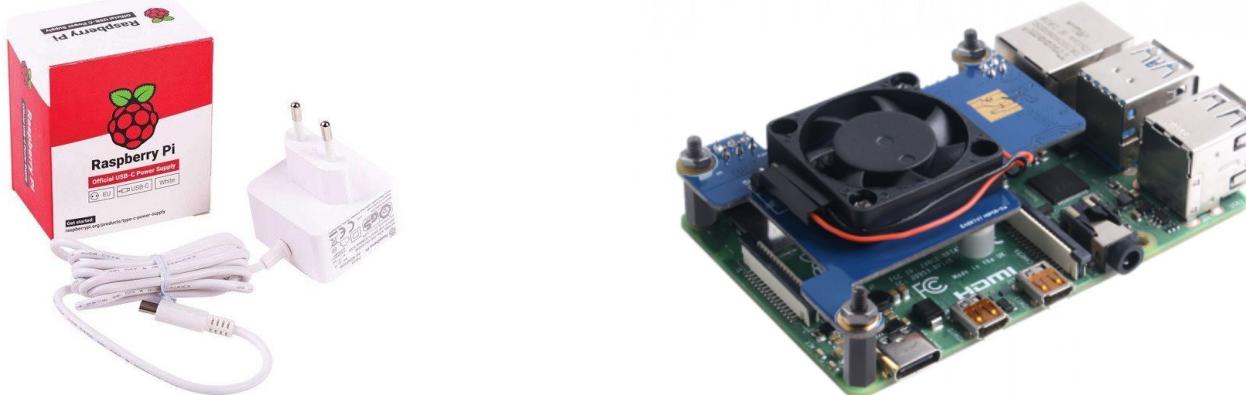
uFL / iPEX

Enclosures



Power supply

Finally we will need a **power supply** that can provide at least 10W (specially if its a Raspberry Pi 4). The official Raspberry Pi power supply is perfect. Optionally you might want to use **PoE** (Power Over Ethernet), especially if you plan to use Ethernet as a backhaul. At RAK we also have a great PoE hat you can use for this project.



<https://www.seeedstudio.com/Power-Over-Ethernet-PoE-HAT-for-Raspberry-Pi-4B-3B-p-4391.html>

Which one is the best gateway for me?

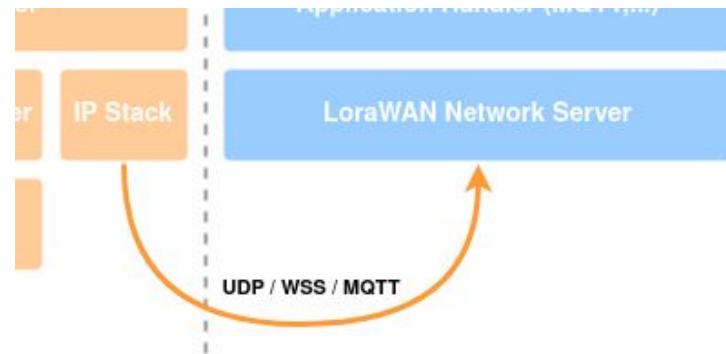
These are the critical key factors for your gateway:

- Convenience (already available parts, form factor, price, backhaul options...)
- Software maintenance and remote management
- Concentrator hardware (same chips, same base design)
- Antenna (frequency adaptation, reflection, frequency bandwidth, gain, short or no cables)
- Location (good view, no obstacles, certain high, ground connection)

What is a LoRa Packet Forwarder?

A **LoRa packet forwarder** is a service (software) running on the host of a LoRa-based gateway (with or without GPS). It forwards RF packets received by the concentrator (uplinks) to a *LoRaWAN Network Server* (LNS) through an IP link. It also transmits RF packets sent by the LNS (downlinks) through the same link to one or multiple devices.

Additionally, it may transmit beacon signals used for time coordinating devices within the network. These beacons may be transmitted GPS-synchronously across the entire network.



Semtech UDP packet forwarder

- This is the original LoRaWAN gateway packet forward protocol, developed as a proof of concept.
- **No built-in authentication or encryption.**
- Relies on a simple IP **UDP** protocol to one port.
- Makes it difficult to cross some firewalls and NAT
- No standardized gateway management options

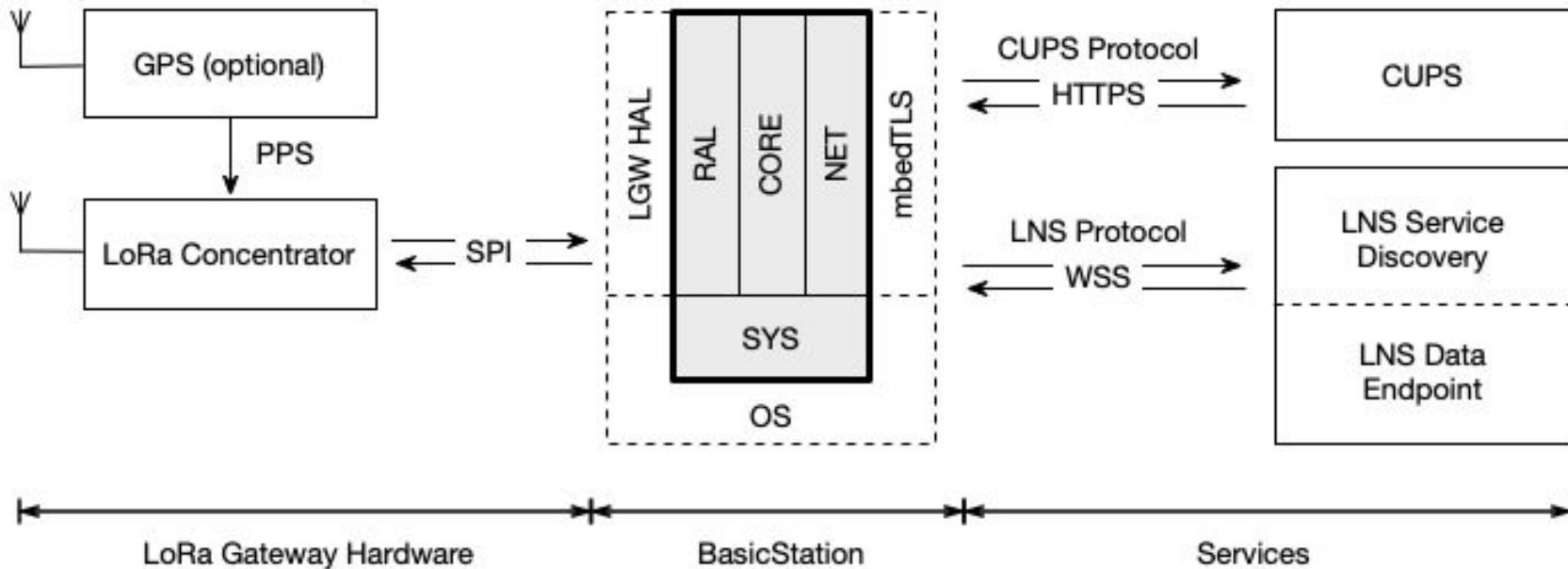
Poly packet forwarder

- UDP packet forwarder sending messages to different LNS at the same time

Multiprotocol packet forwarder

- Same as the poly packet forwarder but includes support for TTN Gateway Connector protocol over MQTT(S).

BasicStation – System Overview



To use the LNS backend protocol to connect to a LoRaWAN Network Server using Basics Station we will need:

- The **URL** of the LNS entry point
- A root or intermediate **certificate** to validate the communication
- A gateway **key** generated by the LNS to authorise the gateway connection

We will be using these parameters shortly in the hands-on part of this workshop...

What is a LoRaWAN Network Server?

A **LoRaWAN Network Server (LNS)** is the service (software) at the core of every LoRaWAN Network that enables connectivity, management, and monitoring of devices, gateways and end-user applications. It is responsible for the security, scalability and reliability of data routing throughout the network.

There are several LNS available as SaaS from different providers and also a few Open Source solutions. **The Things Stack** and **ChirpStack** are two of the most advanced open-source LNS solutions.



**THE THINGS
STACK**



Low-power transmission = Long autonomy (in years)

Long Range (wide area) = Low-cost network (with low-cost subscriptions)

LPWAN networks have a common architecture:

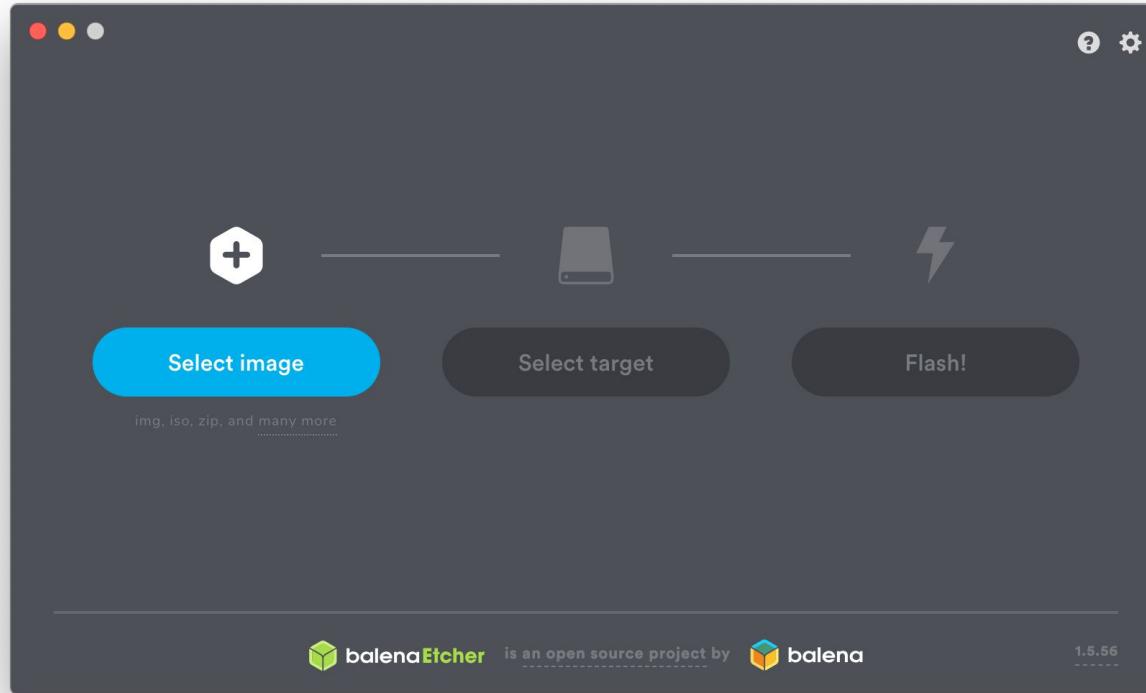
- The devices messages are captured by multiple antennas around.
- The antennas forward the messages to a network server owned by the network operator (private or public)
- Then the network server transfers the payload to the custom backend, eventually, roam it to another network server.

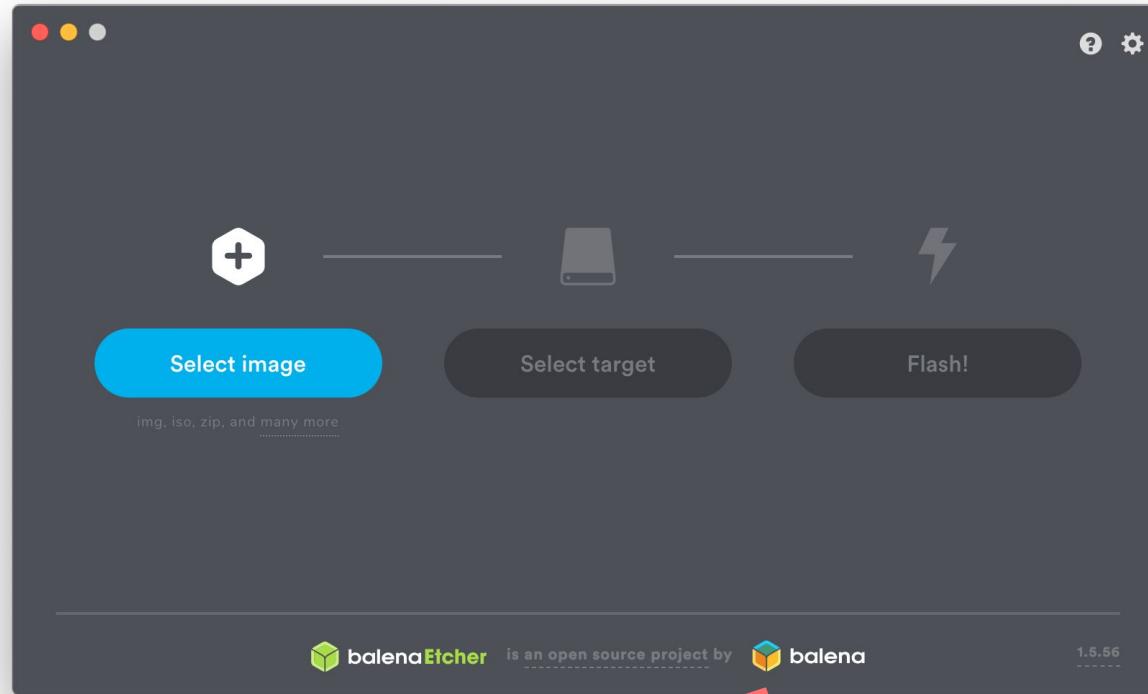
How to deploy a LoRaWAN gateway?

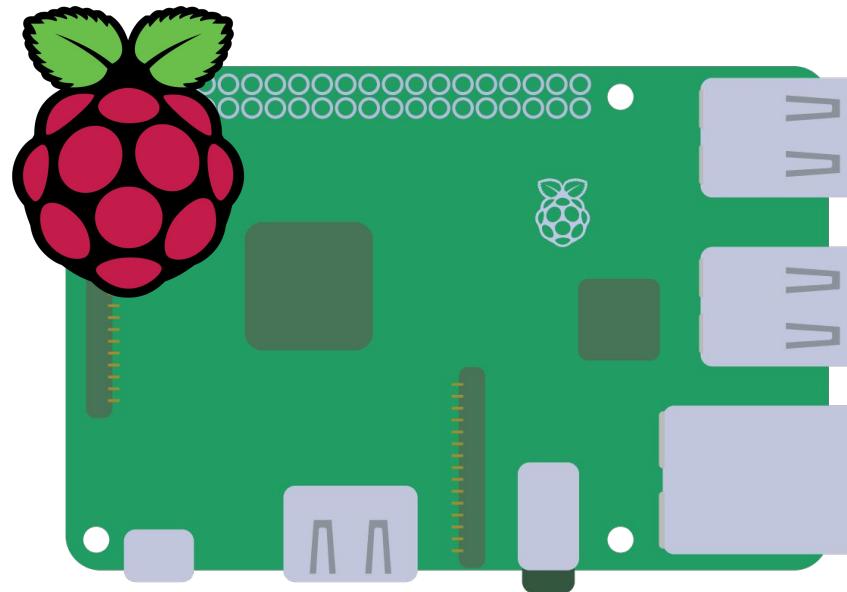


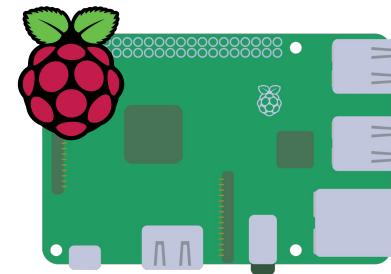
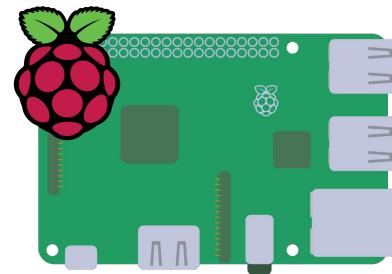
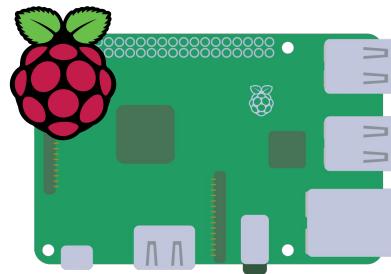
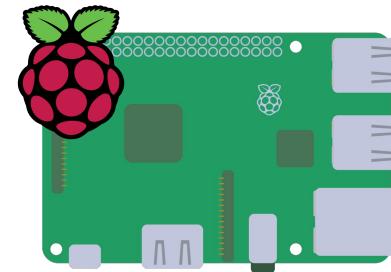
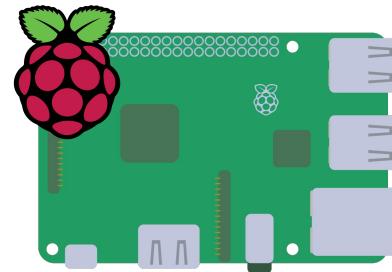
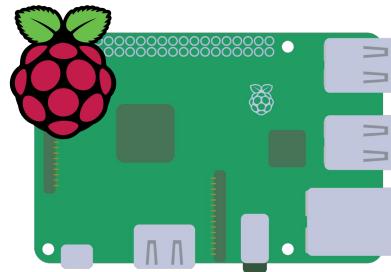
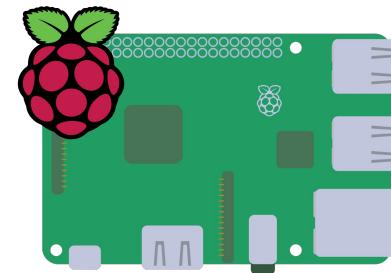
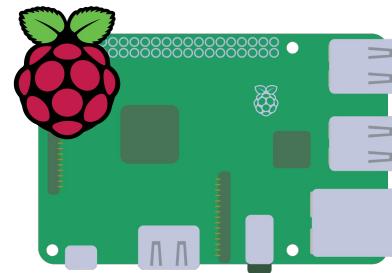
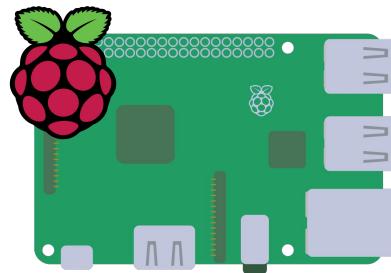
DISCLAIMER: I work at  balena

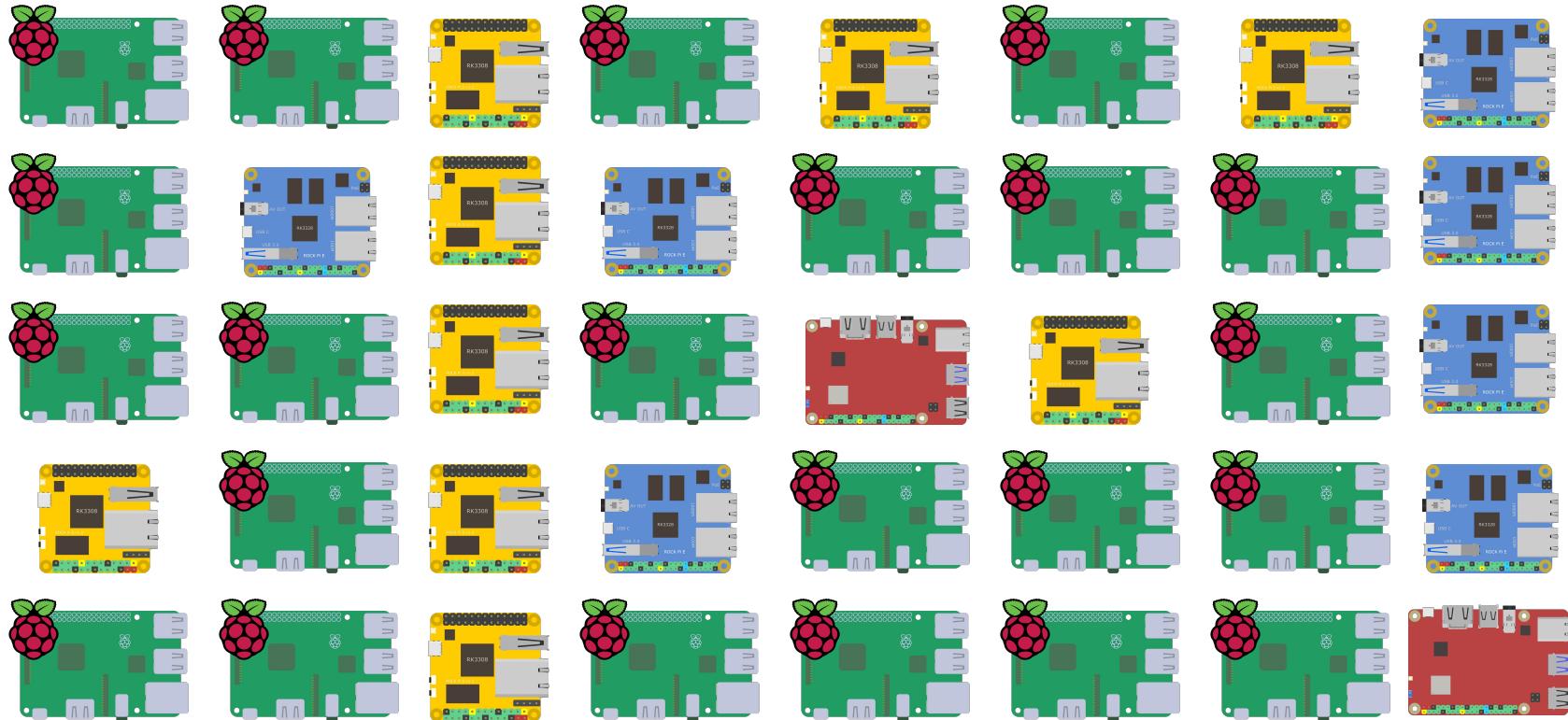
Anyone know this?











Fleet management

dashboard.balena-cloud.com

David Tischler DT

Applications > DemoApplication

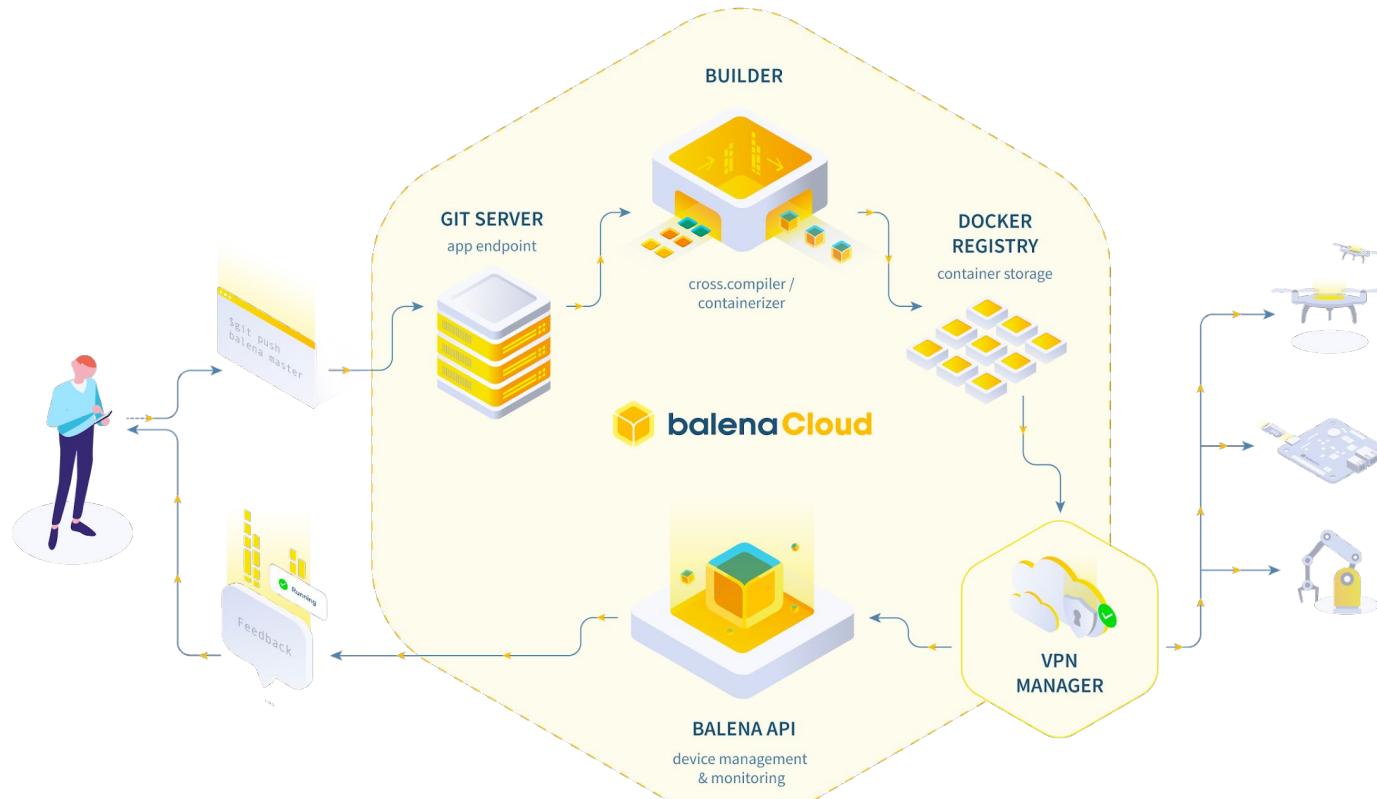
Fleet Location

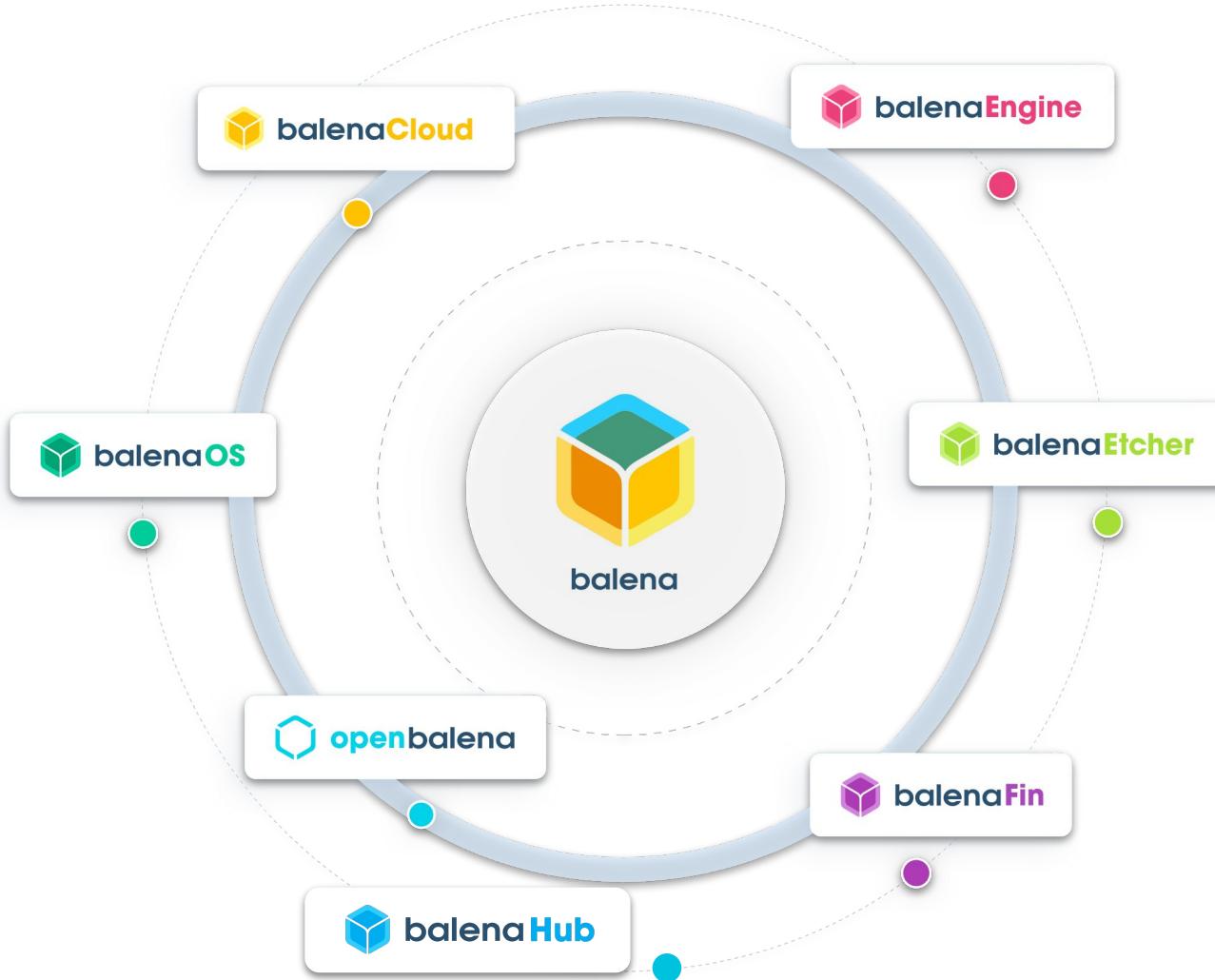
The screenshot shows a fleet management interface for a "DemoApplication" on the balenaCloud dashboard. The left sidebar has a dark blue header with the balenaCloud logo and navigation links: Home, Devices, Fleet Configuration, Environment Variables, Service Variables, Releases, Location (which is highlighted in dark blue), and Members. The main content area is titled "Fleet Location" and displays a world map with yellow dots indicating device locations. Labels on the map include: Greenland, Iceland, Sweden, Norway, Finland, Russia, Mongolia, Japan, South Korea, China, India, Thailand, Pakistan, Saudi Arabia, Iran, Iraq, Turkey, Afghanistan, Algeria, Libya, Egypt, Sudan, Niger, Chad, Nigeria, DRC, Tanzania, Indonesia, Australia, New Zealand, Chile, Argentina, Peru, Bolivia, Venezuela, Colombia, Brazil, Mexico, United States, Canada, and the North Pacific, North Atlantic, South Pacific, South Atlantic, Indian, and Southern Oceans.

Google

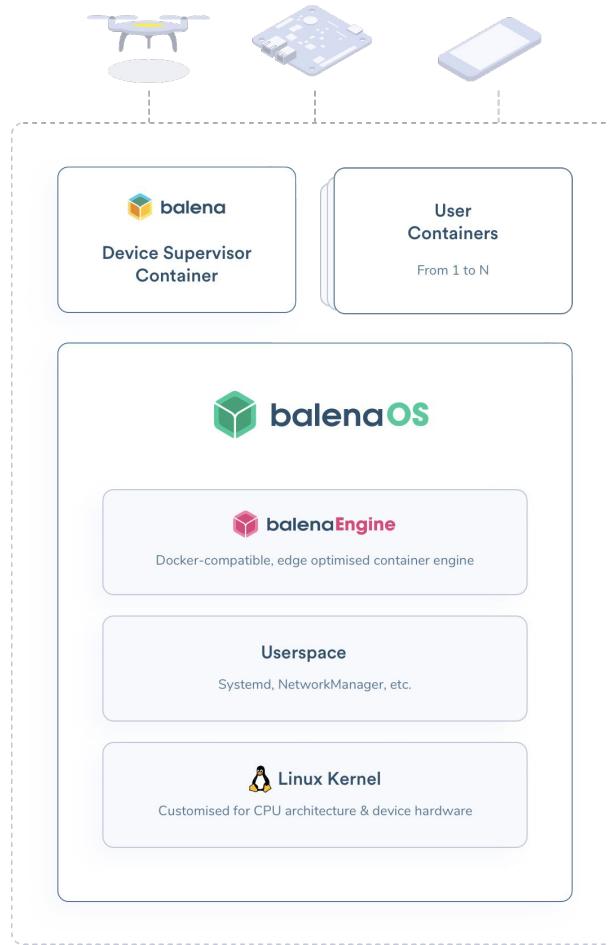
Map data ©2020 | Terms of Use

Software update as a service

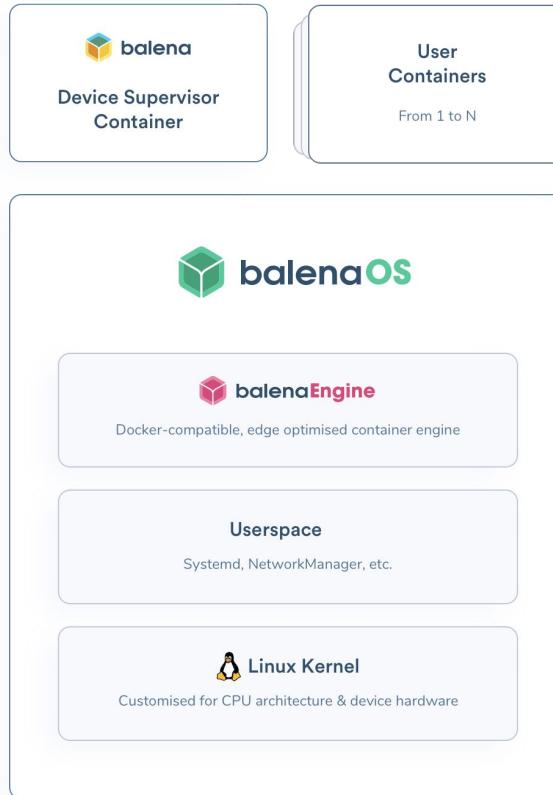




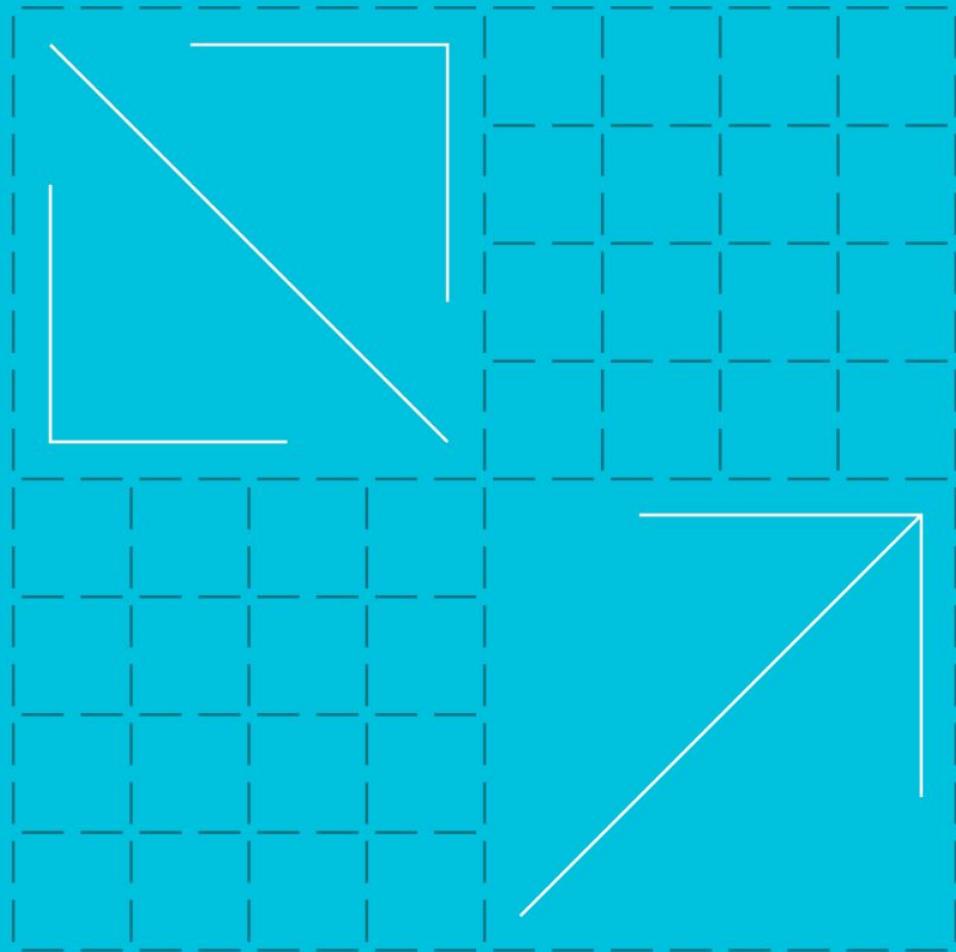
How does it work balena?



Take the benefit of running containers in the edge



**Let's start
“coding” ...**



Hardware

- Raspberry Pi 3 or 4
- LoRa concentrator (e.g. Seeed [WM1302](#) or similar)
- An antenna connected to the LoRa concentrator
- SD card
- Power Supply
- Ethernet Cable or WiFi Access Point for Internet connectivity.
- [SenseCAP S2104](#) (if you have another LoRa node, bring it!)

Software

- [balenaEtcher](#)
- [balenaCloud](#) account
- SenseCAP mobile application (if you use the SenseCAP 210x)

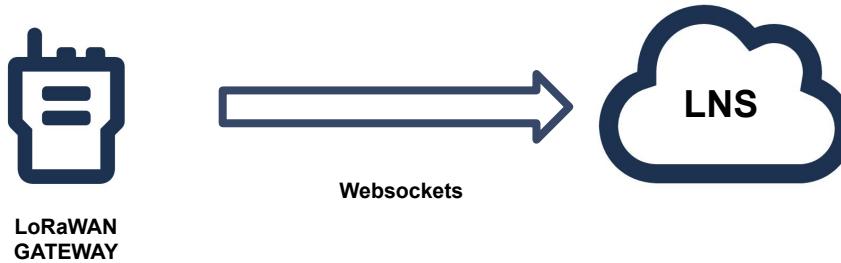
What we are going to build today?

-  Build a LoRaWAN Basics™ Station gateway with the Raspberry Pi, Seeed WM1302 concentrator, balena and The Things Stack LNS
-  Connect a LoRa node to The Things Stack LNS
-  Build your own private LoRaWAN with The Things Stack taking the advantage of the edge with MING.
-  Use the MING stack to visualize data from the LoRa Nodes in the edge.

What we are going to build today?



Build a LoRaWAN Basics™ Station gateway with the Raspberry Pi, Seeed WM1302 concentrator, balena and The Things Stack LNS



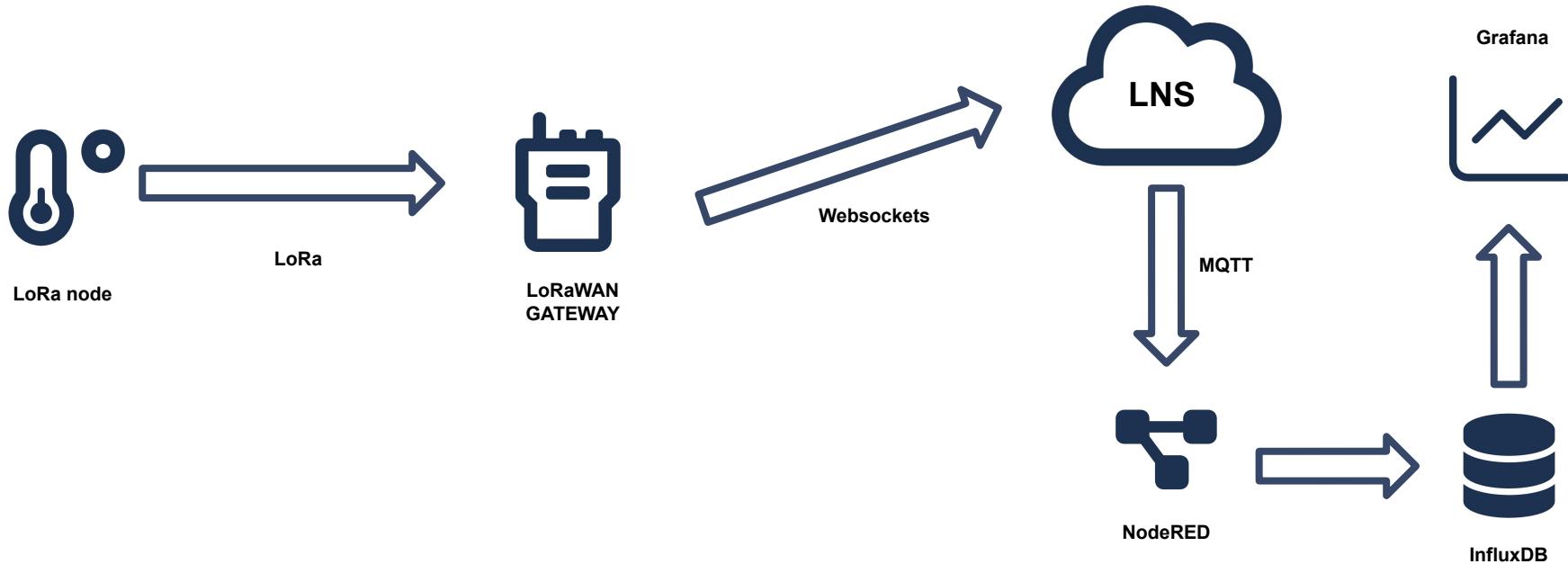
What we are going to build today?

- + Build a LoRaWAN Basics™ Station gateway with the Raspberry Pi, Seeed WM1302 concentrator, balena and The Things Stack LNS
- + Connect a LoRa node to The Things Stack LNS



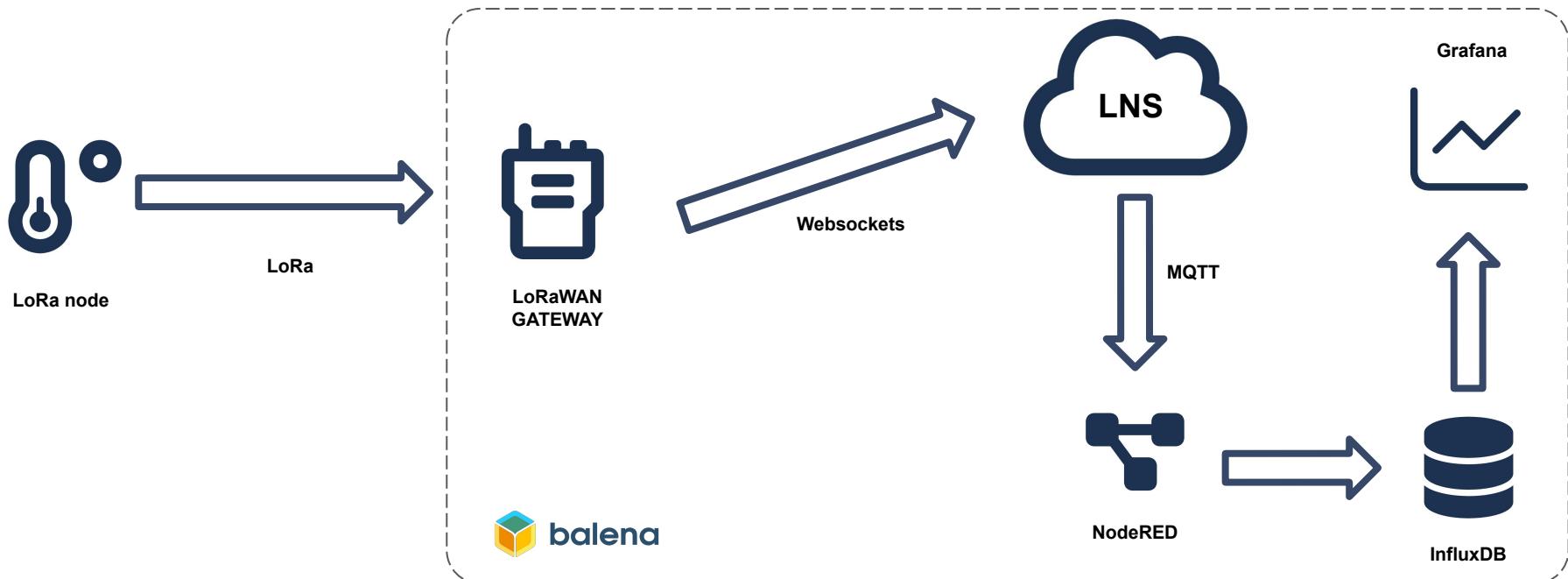
What we are going to build today?

- + Build a LoRaWAN Basics™ Station gateway with the Raspberry Pi, Seeed WM1302 concentrator, balena and The Things Stack LNS. And connect a LoRa node to The Things Stack LNS.
- + Build your own private LoRaWAN with The Things Stack taking the advantage of the edge with MING.



What we are going to build today?

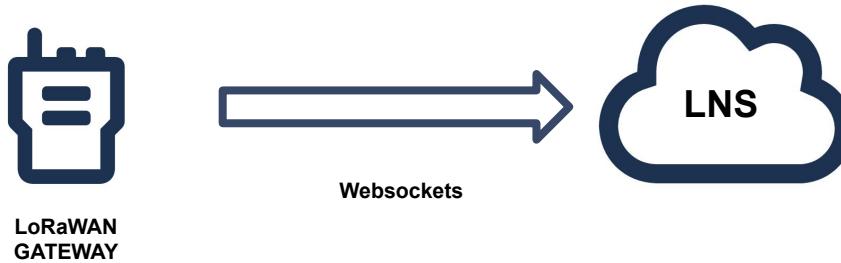
- + Build a LoRaWAN Basics™ Station gateway with the Raspberry Pi, Seeed WM1302 concentrator, balena and The Things Stack LNS. And connect a LoRa node to The Things Stack LNS.
- + Build your own private LoRaWAN with The Things Stack taking the advantage of the edge with MING.



What we are going to build today?



Build a LoRaWAN Basics™ Station gateway with the Raspberry Pi, Seeed WM1302 concentrator, balena and The Things Stack LNS



Go to hub.balena.io to find the App to deploy

arm
DEVSUMMIT

The screenshot shows the balenaHub website interface. On the left, a sidebar has a 'Fleets' tab selected, followed by 'Apps', 'Blocks', 'Organizations', and 'Device Types'. The main area displays a grid of app cards:

- balena-rpiplay** by rahul-thakoor: Turn a Raspberry Pi into an Airplay server using RPiPlay to enable screen mirroring on tvs, monitors and projectors. Works with 2 more... v0.3.1.
- screenly-ose-pi3** by screenly-ose: Updated. The most popular digital signage project on GitHub. Works with 1 more... v0.0.0+rev13.
- HELIUM-RAK** by NebraLtd: Manage your RAK v1.5, v2 or MNTD Helium Miner remotely using Nebra and balenaCloud. Works with 1 more... v0.0.1+rev37.
- rosetta-at-home-arm** by balenaLabs: Fold for Covid - Help fight the COVID-19 pandemic with your old laptop, Raspberry Pi, or other spare computer. Works with 2 more... v0.6.5.
- balenaSound** by balenaLabs: Build a single or multi-room streamer for an existing audio device using a Raspberry Pi! Supports Bluetooth, Airplay and Spotify Connect. Works with 1 more... v3.9.2.
- Pi-hole** by gh_klutchell's Organization: Pi-hole is a Linux network-level advertisement and Internet tracker blocking application! Works with 2 more... v2022.10.0+rev1.
- chi-edge-workers** by Chameleon: Enroll your edge / IoT device into the CHI@Edge testbed for computer science research. Note: you must be associated with a Chameleon allocation... Works with 4 more... v0.1.0+rev13.
- screenly-ose-pi4** by screenly-ose: Updated. The most popular digital signage project on GitHub. Works with 1 more... v0.0.0+rev13.
- pety** by Coolab: Pety uses Pi-Hole to schedule Internet blocks, sync...
- alpha** by Samizdapp: Updated. No description.
- balena-sound** by g_ema_alvarez's Organization: Build a single or multi-room streamer for an...
- Moultimmediat** by stephane_bettbeder's Organization: No description.

A vertical 'Send feedback' button is located on the right side of the page.

Fork this fleet of “standalone-lorawan-gw” or “basicstation” application

arm
DEVSUMMIT

The screenshot shows the balenaHub website interface. On the left, a sidebar navigation bar includes links for 'Organizations', 'xoseperez', 'Fleets', and 'standalone lorawan gw'. The main content area displays a map titled 'standalone-lorawan-gw' by Xose Pérez. The map shows a rural landscape with roads, rivers, and various locations like Bethlehem Church, Santa's Wonderland, and Millican Reserve. A callout box on the left side of the map provides information about the fleet: 'Get standalone-lorawan-gw', 'It's free and open source', '1 Active devices, last 28 days', and a 'Get started' button. Below the map, there is a 'Description' section stating 'Deploys TTS Stack, BasicStation and other tools for a standalone LoRaWAN network solution.' There are also sections for 'Works With' (showing icons for Raspberry Pi, BeagleBoard, and other components), 'Release notes' (version v0.0.0+rev113 | 22 Sep 2022, with a note 'No release notes'), and a 'More from Xose Pérez' section. At the bottom right of the map area, there are buttons for 'Fork this fleet' and 'Report issue'. The URL in the browser address bar is <https://hub.balena.io/organizations/xoseperez/fleets/standalone-lorawan-gw>.

You also can go to [the repo](#) of the app and deploy it from there or fork it and modify the source code

The screenshot shows a GitHub repository page for [xoseperez/standalone-lorawan-gateway-balena](https://github.com/xoseperez/standalone-lorawan-gateway-balena). The repository has 5 unwatched stars, 6 forks, and 12 starred users. It contains 2 branches and 0 tags. The master branch has 34 commits, with the latest being a merge branch 'master' of github.com/xoseperez/balena-tts-lns at commit a64fb2 on 20 Sep. The repository description is: "Deploys the The Things Stack LoRaWAN Network Server Open Source Edition, Basics™ Station and Backend tools using Balena." It includes tags for raspberry-pi, influxdb, node-red, grafana, tts, lora, lorawan, lorawan-gateway, ttn, the-things-network, ins, balena, basicsation, lorawan-network-server, and the-things-stack. The README.md file describes the Standalone LoRaWAN Gateway, mentioning it deploys the The Things Stack LoRaWAN Network Server (Open Source Edition), Basics™ Station packet forwarder and other services using Docker or Balena.io. It runs on a PC, a Raspberry Pi 3/4, Compute Module 3/4 or balenaFin with SX1301, SX1302 or SX1303 LoRa concentrators (e.g. RAK831, RAK833, RAK2245, RAK2247, RAK2287, RAK5146, Seeed WM1302 and IMST IC880a among others). It is a Work In Progress and intended for local (LAN) deployments. The Introduction section discusses deploying a Standalone LoRaWAN Gateway using Basics™ Station Semtech Packet Forwarder and The Things Stack LoRaWAN Network Server in a docker container or as a balena.io fleet. The Main features section lists support for AMD64 (x86_64), ARMv8 and ARMv7 architectures, support for SX1301, SX1302, EX1303 and SX1308 concentrators, Timeseries database and dashboard applications included, and almost one click deploy and highly configurable. Components used include The Things Stack, Basics™ Station, and Node-RED.

“Deploy with balena” or “Fork this fleet” buttons are going to deploy that project into balenaCloud

arm
DEVSUMMIT



Deploy with balena



Fork this fleet

If you don't have balenaCloud account, just sign up. It's free up to 10 devices forever!

arm
DEVSUMMIT

The screenshot shows a web browser window for the 'balena dashboard | Login' at dashboard.balena-cloud.com/login. The page features the balenaCloud logo and a message stating 'Your first ten devices are fully-featured and free'. A callout bubble on the left side promotes a 'Standalone LoRaWAN Gateway' project by user 'xoseperez', which deploys TTS Stack, BasicStation, and other tools for a standalone LoRaWAN network solution. The main right side contains a 'Log in' form with fields for Email* and Password*, and buttons for Log in, Log in with GitHub, and Log in with Google. There are also links for 'Forgot password?' and 'Need help?'. A small 'Need help?' icon is located in the bottom right corner.

balenaCloud

Your first ten devices are fully-featured and free

To continue, login to balenaCloud

Standalone LoRaWAN Gateway

by xoseperez

Deploys TTS Stack, BasicStation and other tools for a standalone LoRaWAN network solution.

Log in

New to balena? Sign up for free.

Log in with GitHub Log in with Google

or log in with

Email*

Password*

Log in

Forgot password?

Need help?

Create a fleet and you will see the App being released on your fleet. Click “Add Device”

arm
DEVSUMMIT

The screenshot shows the balena dashboard interface. On the left, a sidebar navigation includes: Organizations, Marc Pous, Fleets, standalone-lorawan-gateway-arm-dev-summit-2022 (selected), Summary, Devices, Releases, Variables, Configuration, Settings, Provisioning Keys, Members, and Teams. At the bottom left is a Changelog link (v15.15.13). The main content area displays a fleet named "standalone-lorawan-gateway-arm-dev-summit-2022". The fleet summary shows: Architecture (aarch64), Slug (marc6/standalone-lorawan-gateway-arm-dev-summit-2022), Created (Sep 23rd 2022, 2:17 pm), and Microservices. Below this is a table of devices:

Name	Status	Device type	Last seen	Created on	UUID	OS version	OS line	OS variant	Supervisor version	Support access	IP address	Actions
Marc-pi4	Online	Raspberry Pi 4 (using 64bit OS)	Online (about 17 hours)	Oct 5th 2022, 12:43 AM	b9fbcd4	balenaOS 2.105.1+rev1	Development		14.2.0	Off	192.168.1.46	7
muddy-pie	Offline	Raspberry Pi 4 (using 64bit OS)	about 1 month ago	Sep 19th 2022, 10:27 PM	f723db7	balenaOS 2.103.1+rev1	Development		14.0.14	Off	10.0.89.101	8
icy-time	Offline	Raspberry Pi CM4 IO Board	about 1 month ago	Sep 23rd 2022, 2:33 PM	bbcb4a9	balenaOS 2.94.4+rev1	Production		12.11.36	Off	10.42.0.242	7

On the right, there are sections for Devices (3 devices online) and Releases (1 release, titled "track latest"). A "Create release" button is available. The bottom right corner has a "Help" link.

Download the balenaOS image with your networking credentials and Flash the image with Etcher.

arm
DEVSUMMIT

The screenshot shows the balena dashboard interface. On the left, there's a sidebar with navigation links like 'Organizations', 'Marc Pous', 'Fleets', 'standalone-lorawan...', 'Summary', 'Devices', 'Releases', 'Variables', 'Configuration', 'Provisioning Keys', 'Members', and 'Teams'. Below the sidebar, there's a 'Changelog' section with 'v15.15.20'.

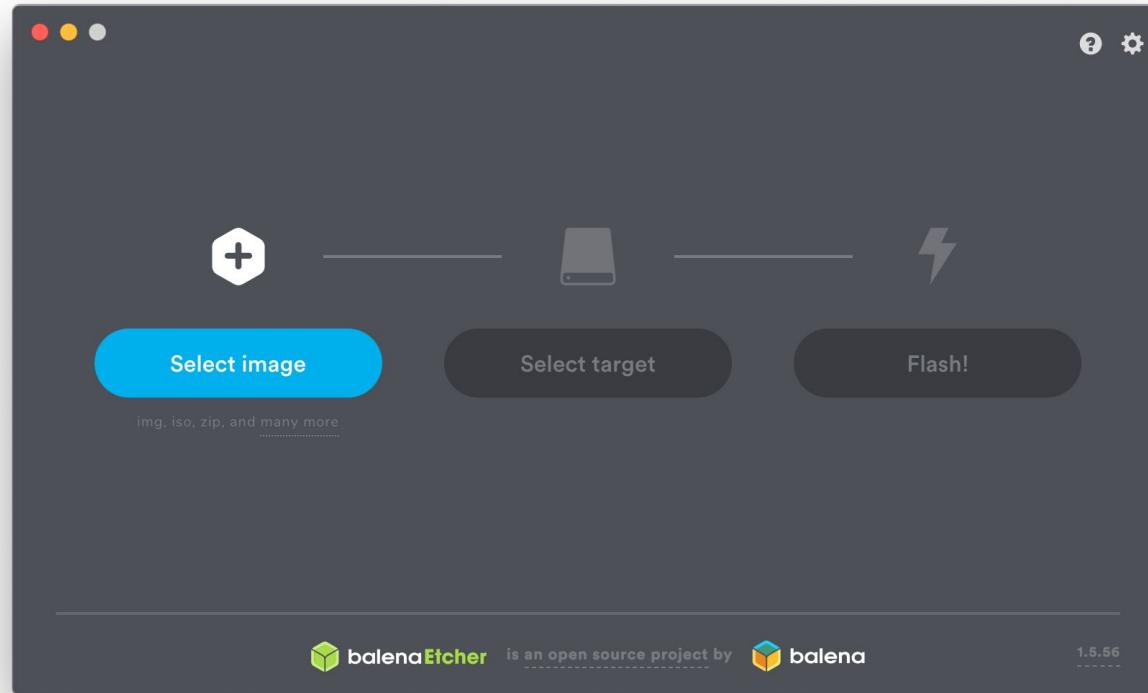
The main area displays a fleet named 'standalone-lorawan-gateway-arm-dev-summit-2022'. It shows a list of devices: 'Marc-pi4' (online), 'alan-pi3-us' (online), 'muddy pie' (offline), 'icy-time' (offline), and 'Seeded Pi' (offline). A summary bar indicates 5 devices online and 0 offline.

A central modal window titled 'Add new device' is open. It contains fields for 'Select device type' (set to 'Raspberry Pi 4 (using 64bit OS)'), 'Select OS type' (set to 'balenaOS ESR'), and 'Select version' (set to 'v2022.10.0 (next, recommended)'). There are also sections for 'Instructions' (with steps for configuration, download, and physical setup), 'Development' (warning about local mode), 'Production' (warning about easy access), and 'Network Connection' (set to 'Ethernet only'). At the bottom of the modal is a large blue 'Flash' button.

To the right of the modal, there's a 'Devices' section with a status bar (5 online, 0 offline) and a 'Releases' section showing a list of releases with columns for IP address, Public address, and MAC address.

Once the SD card is successfully flashed, introduce it on your Raspberry Pi and power it up!

arm
DEVSUMMIT



The device will appear on the list, and then you will be able to see the services being installed on it.

The screenshot shows the balena dashboard interface for a device named "Marc-pi4".

Device Summary:

- Status:** Online (UUID: b9fbcd4)
- Online For:** 17 hours
- Supervisor Version:** 14.2.0
- Support Access:** Off
- Local IP Address:** 192.168.1.46
- Public IP Address:** 79.153.119.0
- Tags:** EUI:DCA632FFE554B34, URL: https://balena.io
- Notes:** EUI:DCA632FFE554B34

Services:

Service	Status	Release
basicstation	Running	d3542ad
grafana	Running	d3542ad
influxdb	Running	d3542ad
node-red	Running	d3542ad
postgres	Running	d3542ad
redis	Running	d3542ad
stack	Running	d3542ad
wifi-connect	Running	d3542ad

Metrics:

- CPU: -80%
- Temperature: -71C
- Memory: 981 MB / 3.8 GB
- Storage: 2.7 GB / 28.3 GB

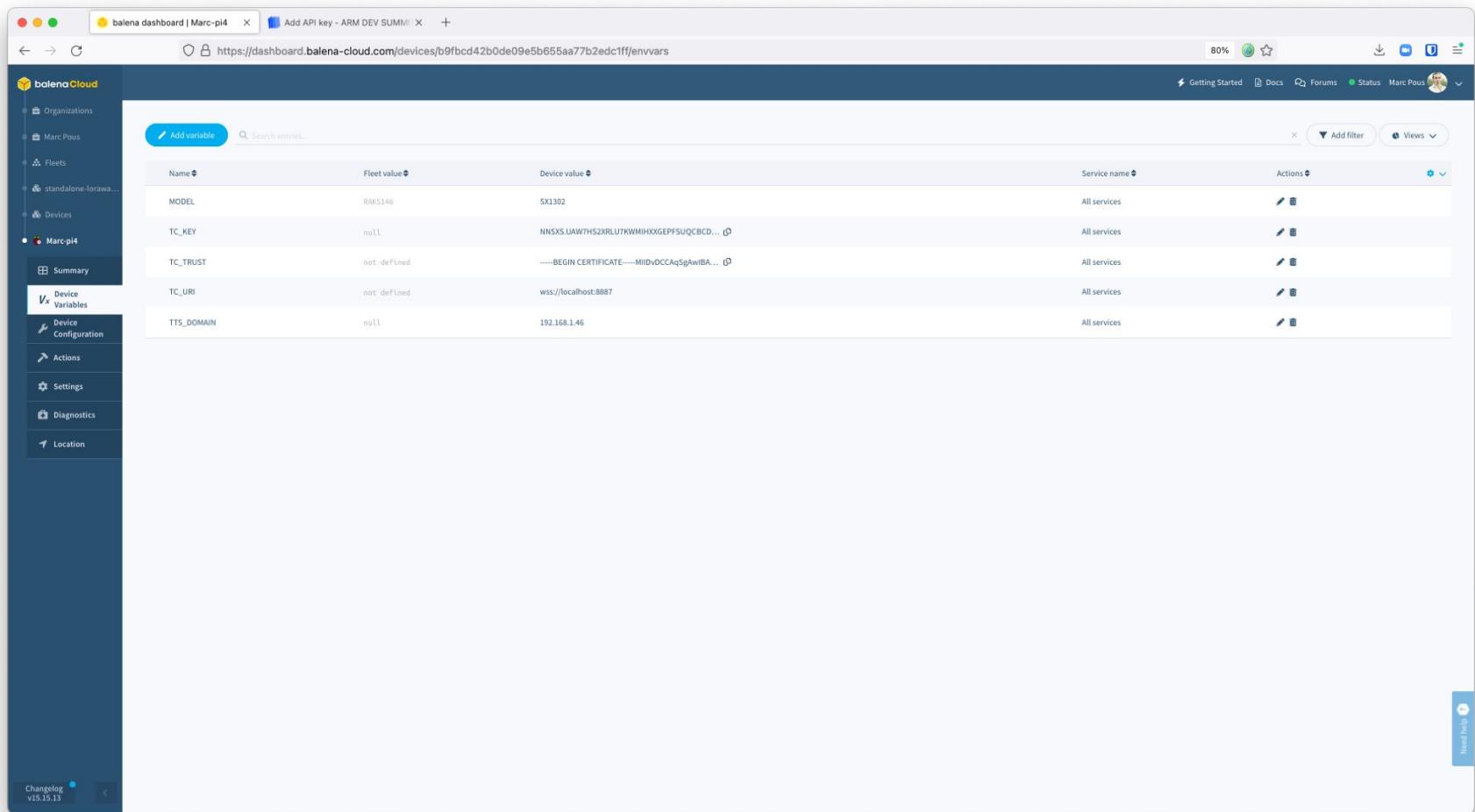
Logs:

```
05.11.22 08:11:57 (+0000) stack INFO Client error {"duration": 0.0068, "error": "error:pkg/auth:token (invalid token)", "http.method": "GET", "http.path": "/traffic/eui-DCA632FFE554B34", "http.status": 400, "namespace": "web", "peer.address": "172.18.0.1:65144", "peer.real_ip": "172.18.0.1", "request_id": "01GH30HCS54DAQH94BTSNGVGF5"}  
05.11.22 08:11:57 (+0000) basicstation 2022-11-05 08:11:57.611 [AIOD:ERRO] [4] WS upgrade failed with HTTP status code: 400  
05.11.22 08:11:57 (+0000) basicstation 2022-11-05 08:11:57.611 [AIOD:ERBU] [4] WS connection shutdown...  
05.11.22 08:11:57 (+0000) basicstation 2022-11-05 08:11:57.612 [TC:VERB] Connection to MUXS closed in state 3  
05.11.22 08:11:57 (+0000) basicstation 2022-11-05 08:11:57.612 [TC:INFO] INFOs reconnect backoff 10s (retry 1)  
05.11.22 08:12:02 (+0000) stack INFO Request handled ("duration": 0.0001, "http.method": "GET", "http.path": "/health/live", "http.status": 200, "namespace": "web", "peer.address": "127.0.0.1:56262", "request_id": "01GH30HFP8A44 STCD005PW34T3")  
05.11.22 08:12:04 (+0000) influxdb [httpd] [72.18.0.4 - - [05/Nov/2022:08:12:04 +0000] "GET /query?db=balena&q=showfield+keys HTTP/1.1" 200 70 "-" "Python-urllib/3.7" 88643c4e-5ce1-11ed-8814-024ac120005 1140  
05.11.22 08:12:04 (+0000) grafana Interim dashboard sync skipped: No schema found.
```

Terminal:

Select a target ... Start terminal session

Copy your local IP address and override the variable `TTS_DOMAIN`.



The screenshot shows the balena dashboard interface for a device named 'Marc-pi4'. The left sidebar contains navigation links for Organizations, Fleets, standalone-lorawan..., Devices, Summary, Device Variables, Device Configuration, Actions, Settings, Diagnostics, and Location. The main content area displays a table of environment variables:

Name	Fleet value	Device value	Service name	Actions
MODEL	RASPi4G	SX1302	All services	Edit Delete
TC_KEY	null	NNSXS.UAW7HS2XRLU7KWMIHGXGEPFSUQCBCD... ⓘ	All services	Edit Delete
TC_TRUST	not_defined	-----BEGIN CERTIFICATE-----MIIDvOCQAqSgAwIBA... ⓘ	All services	Edit Delete
TC_URI	not_defined	wss://localhost:8887	All services	Edit Delete
TTS_DOMAIN	null	192.168.1.46	All services	Edit Delete

The URL in the browser bar is https://dashboard.balena-cloud.com/devices/b9fbcd42b0de09e5b655aa77b2edc1ff/envvars. The status bar at the bottom left shows 'Changelog v15.15.13' and the bottom right has a 'Need Help?' button.

Put `https://<your local ip address>` on your browser and accept the risks of the certificate.

The screenshot shows a Firefox browser window with the following details:

- Title Bar:** balena dashboard | Marc-pi4
- Address Bar:** https://192.168.1.46
- Content Area:**
 - Warning Icon:** An orange exclamation mark icon.
 - Section Title:** Avís: Risc potencial de seguretat
 - Description:** El Firefox ha detectat una amenaça potencial de seguretat i ha interromput la connexió a 192.168.1.46. Si visiteu aquest lloc, els atacants podrien robar informació com ara contrasenyes, correus electrònics o detalls de targetes de crèdit.
 - Section Title:** Què hi podeu fer?
 - Description:** Molt probablement, l'error és del lloc web i no hi podeu fer res per resoldre'l.
 - Description:** Si esteu en una xarxa empresarial o utilieu programari antivirus, podeu posar-vos en contacte amb l'equip d'assistència tècnica. També podeu notificar el problema a l'administrador del lloc web.
 - Link:** [Més informació...](#)
- Buttons:**
 - Vés enrere (recomanat)
 - Avançat...

Login into The Things Stack with User ID “admin” and “changeme” as a password

arm
DEVSUMMIT

The image shows a split-screen view. On the left, a web browser window displays the 'Login - Account - The Things Stack' page. The URL in the address bar is https://192.168.1.46/oauth/login?n=%2Foauth%2Fauthorize%3Fclient_id%3Dconsole%26redirect_uri%3D%252Fconsole%252Foauth%252Fcallback%26response_type%3Dcode%26state%3DsSsDVaXjuNGuvT. The page header includes 'THE THINGS STACK Open Source'. The main content is titled 'The Things Stack for LoRaWAN Account' with a message 'Please login to continue'. It features input fields for 'User ID*' and 'Password*', both marked with red asterisks. Below the fields are 'Login' and 'Create an account' buttons, and a 'Forgot password?' link. At the bottom, it says '© 2022 The Things Stack by The Things Network and The Things Industries'. On the right side of the image is a large, stylized map composed of blue hexagonal tiles. Each tile contains a white icon representing a different type of IoT device or service, such as a house, bus, puzzle piece, phone, lock, ship, eye, Wi-Fi signal, temperature gauge, speech bubble, and checkmark.

Once successfully logged in, go to “Register a Gateway”

The screenshot shows a web browser window for the 'balena dashboard | Marc-pi4' at 'https://192.168.1.46/console/'. The title bar says 'Overview - Console - The Thing'. The main content is the 'THE THINGS STACK Open Source' console. At the top, there are navigation links: Overview (selected), Applications, Gateways, and Organizations. On the right, there is a user profile for 'admin'. The central area has a heading 'Welcome to the Console!' followed by text: 'Get started right away by creating an application or registering a gateway.' Below this are two large call-to-action boxes: one for 'Create an application' (represented by a blue line-art icon of a computer monitor) and one for 'Register a gateway' (represented by a blue line-art icon of a small device with a Wi-Fi signal). At the bottom, there are links for 'Version info' and 'Component status'.

balena dashboard | Marc-pi4 Overview - Console - The Thing https://192.168.1.46/console/ THE THINGS STACK Open Source Overview Applications Gateways Organizations admin Welcome to the Console! Get started right away by creating an application or registering a gateway. Need help? Have a look at our Documentation. Create an application Register a gateway Version info Component status

Copy the TAG EUI from the balenaCloud dashboard. Be careful and just copy the EUI

The screenshot shows the balenaCloud dashboard for a device named "Marc-pi4".

Device Summary:

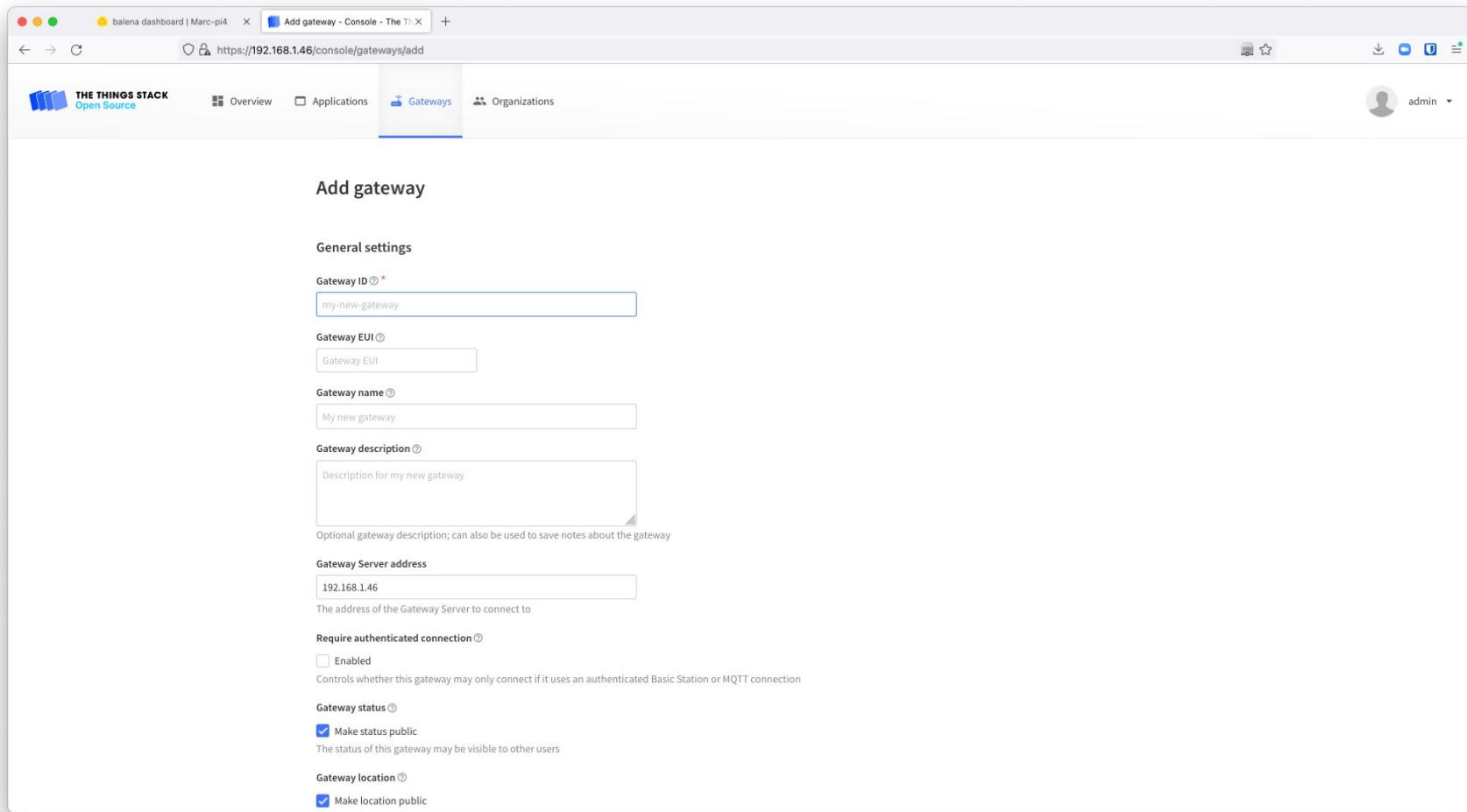
- Status:** Online (UUID: b9fbcd4)
- OS:** Raspberry Pi 4 (using 64bit OS)
- OS VARIANT:** development
- SUPERVISOR VERSION:** 14.2.0
- SUPPORT ACCESS:** Off
- LOCAL IP ADDRESS:** 192.168.1.46
- PUBLIC IP ADDRESS:** 79.153.119.0
- MAC ADDRESS:** DC:A6:32:55:4B:34
- PUBLIC DEVICE URL:** Off (On)

Logs: Shows system logs with entries related to stack, basicstation, influxdb, and grafana.

Services: A table listing running services:

Service	Status	Release
basicstation	Running	d3542ad
grafana	Running	d3542ad
influxdb	Running	d3542ad
node-red	Running	d3542ad
postgres	Running	d3542ad
redis	Running	d3542ad
stack	Running	d3542ad
wifi-connect	Running	d3542ad

Changelog: v15.15.13



The screenshot shows a web browser window titled "Add gateway - Console - The Things Stack". The URL is https://192.168.1.46/console/gateways/add. The page has a header with tabs: Overview, Applications, Gateways (which is selected), and Organizations. On the right, there's a user profile for "admin". The main content area is titled "Add gateway" and contains several input fields under the heading "General settings".

General settings

Gateway ID (required): my-new-gateway

Gateway EUI: Gateway EUI

Gateway name: My new gateway

Gateway description: Description for my new gateway

Optional gateway description; can also be used to save notes about the gateway

Gateway Server address: 192.168.1.46

The address of the Gateway Server to connect to

Require authenticated connection: Enabled

Controls whether this gateway may only connect if it uses an authenticated Basic Station or MQTT connection

Gateway status: Make status public

The status of this gateway may be visible to other users

Gateway location: Make location public

Once successfully created, go to “API keys”

balena dashboard | Marc-pi4 Overview - ARM DEV SUMMIT + https://192.168.1.46/console/gateways/armdevsummit admin

THE THINGS STACK Open Source Overview Applications Gateways Organizations

ARM DEV SUMMIT Overview Live data Location Collaborators API keys General settings

Gateways > ARM DEV SUMMIT

ARM DEV SUMMIT ID: armdevsummit Disconnected 1 Collaborator 0 API keys

General information

Gateway ID	armdevsummit
Gateway EUI	DC A6 32 FF FE 55 4B 34
Gateway description	None
Created at	Nov 7, 2022 00:00:27
Last updated at	Nov 7, 2022 00:00:27
Gateway Server address	192.168.1.46

LoRaWAN information

Frequency plan	EU_863_870_TTN
Global configuration	Download global_conf.json

Live data See all activity → Waiting for events from armdevsummit...

Location Change location settings → No location information available

< Hide sidebar

The screenshot shows a web browser window titled "Add API key - ARM DEV SUMMIT". The URL is <https://192.168.1.46/console/gateways/armdevsummit/api-keys/add>. The page is part of "THE THINGS STACK Open Source" and is under the "ARM DEV SUMMIT" organization. The sidebar on the left includes links for Overview, Applications, Gateways (which is selected), Organizations, and API keys (which is also selected). The main content area shows the "Add API key" form. It has fields for "Name" (containing "My new API key"), "Expiry date" (empty), and "Rights". Under "Rights", the "Grant individual rights" option is selected. The "Link as Gateway to a Gateway Server for traffic exchange, i.e. write uplink and read downlink" checkbox is checked. Other options like "Select all" and various gateway-related permissions are available but not selected.

THE THINGS STACK
Open Source

ARM DEV SUMMIT

Overview Applications Gateways Organizations

admin

Gateways > ARM DEV SUMMIT > API keys > Add

Add API key

Name

Expiry date

Rights *

Grant all current and future rights

Grant individual rights

Select all

Delete gateway

View gateway information

Link as Gateway to a Gateway Server for traffic exchange, i.e. write uplink and read downlink

View gateway location

Retrieve secrets associated with a gateway

View and edit gateway API keys

Edit basic gateway settings

View and edit gateway collaborators

View gateway status

Write downlink gateway traffic

Read gateway traffic

Store secrets for a gateway

Create API key

Paste the API key overriding the `TC_URI` variable

The screenshot shows the balena dashboard interface for a device named 'Marc-pi4'. The left sidebar contains navigation links for Organizations, Fleets, standalone-lorawan..., Devices, Summary, Device Variables, Device Configuration, Actions, Settings, Diagnostics, and Location. The main content area displays a table of environment variables:

Name	Fleet value	Device value	Service name	Actions
MODEL	RASPi4G	SX1302	All services	Edit Delete
TC_KEY	null	NNSXS.UAW7HS2XRLU7KWMIHGXGEPFSUQCBCD... ⓘ	All services	Edit Delete
TC_TRUST	not defined	-----BEGIN CERTIFICATE-----MIIDvOCQAqSgAwIBA... ⓘ	All services	Edit Delete
TC_URI	not defined	wss://localhost:8887	All services	Edit Delete
TTS_DOMAIN	null	192.168.1.46	All services	Edit Delete

The top right corner of the browser window shows a status bar with '80%', a globe icon, and a user profile picture.

The gateway will appear as Connected. If there are already LoRa nodes sending data, it will appear on Live Data

The screenshot shows the balena dashboard interface for the 'ARM DEV SUMMIT' application. The top navigation bar includes tabs for Overview, Applications, Gateways (which is selected), and Organizations. On the left sidebar, there are links for Overview, Live data, Location, Collaborators, API keys, and General settings. The main content area displays the 'ARM DEV SUMMIT' gateway details. The gateway icon is highlighted with a red oval. Below it, the ID 'armdevsummit' is shown. The status indicates 'Last activity 53 seconds ago'. The 'General information' section lists the Gateway ID as 'armdevsummit', Gateway EUI as 'DC A6 32 FF FE 55 4B 34', and Gateway description as 'None'. It also shows the creation date as 'Nov 7, 2022 00:00:27' and the last update date as 'Nov 7, 2022 00:00:27'. The 'Gateway Server address' is set to '192.168.1.46'. The 'LoRaWAN information' section shows the Frequency plan as 'EU_863_870_TTN' and a 'Download global_conf.json' button. To the right, there is a 'Live data' section with a message 'Waiting for events from armdevsummit...' and a 'Location' section showing a world map with the message 'No location information available'.

THE THINGS STACK
Open Source

Overview Applications Gateways Organizations

admin

ARM DEV SUMMIT

Overview Live data Location Collaborators API keys General settings

Gateways > ARM DEV SUMMIT

ARM DEV SUMMIT
ID: armdevsummit

Last activity 53 seconds ago

General information

Gateway ID: armdevsummit

Gateway EUI: DC A6 32 FF FE 55 4B 34

Gateway description: None

Created at: Nov 7, 2022 00:00:27

Last updated at: Nov 7, 2022 00:00:27

Gateway Server address: 192.168.1.46

LoRaWAN information

Frequency plan: EU_863_870_TTN

Global configuration: Download global_conf.json

Live data

See all activity →

Waiting for events from armdevsummit...

Location

Change location settings →

No location information available

< Hide sidebar

What we are going to build today?

- + Build a LoRaWAN Basics™ Station gateway with the Raspberry Pi, Seeed WM1302 concentrator, balena and The Things Stack LNS
- + Connect a LoRa node to The Things Stack LNS



We are going to connect these 2 SenseCAP sensors (S2101 and S2104)

seeed studio



S2101

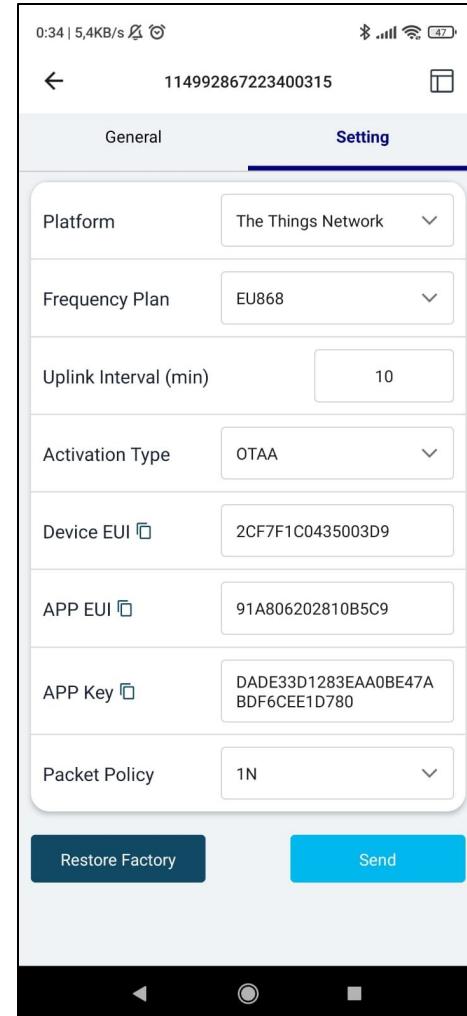
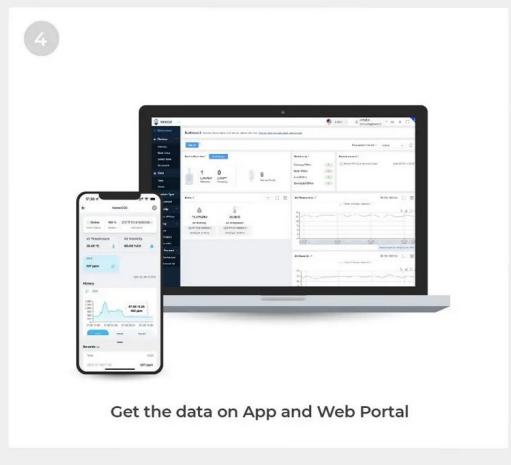
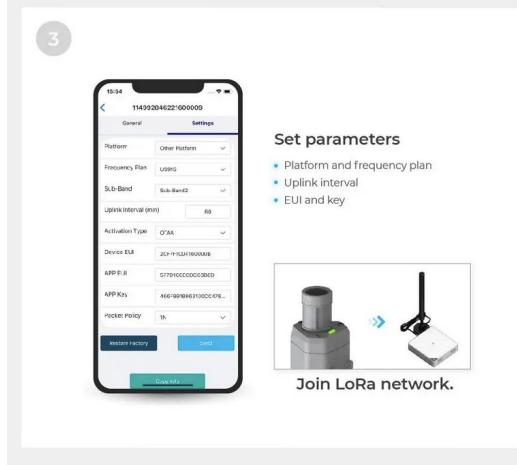
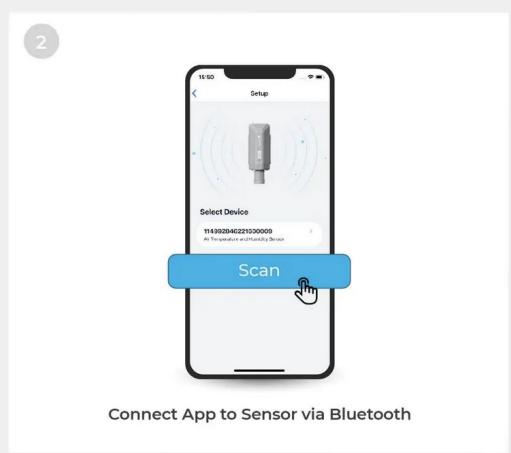


S2104

<https://www.seeedstudio.com/SenseCAP-S2101-LoRaWAN-Air-Temperature-and-Humidity-Sensor-p-5354.html>

<https://www.seeedstudio.com/SenseCAP-S2104-LoRaWAN-Soil-Temperature-and-Moisture-Sensor-p-5357.html>

Install the SenseCAP mate app and change the device settings if needed



Go to The Things Stack and click on Applications

arm
DEVSUMMIT

The screenshot shows a web browser window for 'The Things Stack' application management interface. The URL is <https://192.168.1.46/console/applications>. The top navigation bar includes tabs for 'Overview', 'Applications' (which is selected), 'Gateways', and 'Organizations'. A user profile for 'admin' is visible on the right. The main content area displays a table header for 'Owned applications' with columns: ID, Name, End devices, and Created at. A search bar and a 'Add application' button are also present. Below the table, a message says 'No items found'.

THE THINGS STACK
Open Source

Overview Applications Gateways Organizations

admin

Owned applications

All (Admin) Deleted (Admin)

Search

Add application

No items found

ID Name End devices Created at

© 2022 The Things Stack by The Things Network and The Things Industries

EN v3.21.0 Documentation

Create your Application. Now click on “Add end device”

The screenshot shows the The Things Stack dashboard interface. At the top, there are two tabs: "balena dashboard | Marc-pi4" and "Overview - Temperature - The". The URL in the address bar is <https://192.168.1.46/console/applications/seeded-temperature>. On the right side, there is a user profile for "admin". The main navigation menu includes "Overview", "Applications" (which is selected), "Gateways", and "Organizations". The left sidebar contains links for "Temperature", "Overview", "End devices", "Live data", "Payload formatters", "Integrations", "Collaborators", "API keys", and "General settings". The "End devices" section shows "0 End devices", "1 Collaborator", and "0 API keys". The "General information" section displays the Application ID "seeded-temperature", Created at "Nov 7, 2022 00:05:39", and Last updated at "Nov 7, 2022 00:05:39". The "Live data" section shows a message "Waiting for events from seeded-temperature...". Below this, the "End devices (0)" section includes a search bar, an "Import end devices" button, and an "+ Add end device" button. A table header for "End devices" includes columns for "ID", "Name", "DevEUI", "JoinEUI", and "Last activity". The message "No items found" is displayed below the table. At the bottom left, there is a link "[Hide sidebar](#)". The footer contains copyright information "© 2022 The Things Stack by The Things Network and The Things Industries" and language selection "EN v3.21.0 Documentation".

The screenshot shows a web browser window for 'balena dashboard | Marc-pi4' at the URL <https://192.168.1.46/console/applications/seeed-temperature/devices/add/repository>. The page title is 'Register end device - Temperature'.

The top navigation bar includes links for Overview, Applications (which is selected), Gateways, and Organizations. On the right, there is a user profile for 'admin'.

The left sidebar contains the following menu items:

- Temperature
- Overview
- End devices (selected)
- Live data
- Payload formatters
- Integrations
- Collaborators
- API keys
- General settings

The main content area shows the breadcrumb path: Applications > Temperature > End devices > Register from The LoRaWAN Device Repository.

Register end device

From The LoRaWAN Device Repository [Manually](#)

1. Select the end device

Brand ④ *
 | ↴

Cannot find your exact end device? [Try manual device registration.](#)

2. Enter registration data

Please choose an end device first to proceed with entering registration data

[Register end device](#)

At the bottom left is a link to 'Hide sidebar'. At the bottom right are language and documentation links: EN (selected), v3.21.0, and Documentation.

balena dashboard | Marc-pi4 Register end device - Temperature + https://192.168.1.46/console/applications/seeded-temperature/devices/add/repository Overview Applications Gateways Organizations admin

THE THINGS STACK Open Source

Temperature Overview End devices Live data Payload formatters Integrations Collaborators API keys General settings

Applications > Temperature > End devices > Register from The LoRaWAN Device Repository

Register end device

From The LoRaWAN Device Repository Manually

1. Select the end device

Brand * Model * Hardware Ver. * Firmware Ver. * Profile (Region) *

SenseCAP SenseCAP S2101- L... 1.0 1.0 EU_863_870

SenseCAP S2101- LoRaWAN® Air Temperature and Humidity Sensor
LoRaWAN Specification 1.0.2, RP001 Regional Parameters 1.0.2 revision B, Over the air activation (OTAA), Class A

 SenseCAP 2101 air temperature & humidity sensor satisfies industrial IoT long-distance data acquisition with measuring ranges from -40°C to 85°C and 0 to 100 %RH respectively. With Bluetooth 5.0 for easy wireless configuration and firmware upgrade and a built-in replaceable battery for minimal maintenance, it also supports three different LoRaWAN® network architectures to make it easily become Helium-compatible IoT device.

[Product website](#) | [Data sheet](#)

2. Enter registration data

Frequency plan * Europe 863-870 MHz (SF9 for RX2 - recommended)

AppEUI * 00 00 00 00 00 00 00 00 Fill with zeros

< Hide sidebar

Once the end device is registered, force to send a join (in my case I pushed the button to force the join)

The screenshot shows the balena dashboard interface for managing a LoRaWAN device. The device is identified by the EUI `eui-2cf7f1c0435003d9`. The dashboard displays various configuration tabs: Overview, Applications, Gateways, and Organizations. The Applications tab is selected, showing the device's status under the Temperature application. The General information section provides details like End device ID, Frequency plan, and LoRaWAN version. The Hardware section lists the brand as sensecap and model as sensencaps2101-temp-humid. The Activation information section shows the AppEUI, DevEUI, and AppKey. A world map indicates that no location information is available for this device.

THE THINGS STACK
Open Source

Overview Applications Gateways Organizations admin

Temperature

Overview End devices Live data Payload formatters Integrations Collaborators API keys General settings

eui-2cf7f1c0435003d9
ID: eui-2cf7f1c0435003d9

↑ n/a ↓ n/a • No activity yet ⓘ

Overview Live data Messaging Location Payload formatters Claiming General settings

General information

End device ID	eui-2cf7f1c0435003d9
Frequency plan	Europe 863-870 MHz (SF9 for RX2 - recommended)
LoRaWAN version	LoRaWAN Specification 1.0.2
Regional Parameters version	RP001 Regional Parameters 1.0.2 revision B

Created at: Nov 7, 2022 00:53:24

Live data See all activity →

Waiting for events from eui-2cf7f1c0435003d9...

Hardware

Brand	sensecap
Model	sensencaps2101-temp-humid
Hardware version	1.0
Firmware version	1.0

Activation information

AppEUI	91 A8 06 20 28 10 B5 C9
DevEUI	2C F7 F1 C0 43 50 03 D9
AppKey	*****

Location Change location settings →

No location information available

https://192.168.1.46/console/applications/seeed-temperature/devices/eui-2cf7f1c0435003d9/data

You will see the join request on the Live Data box and the acceptance if everything goes well

THE THINGS STACK
Open Source

Overview Applications Gateways Organizations admin

Temperature Overview End devices Live data Payload formatters Integrations Collaborators API keys General settings

Applications > Temperature > End devices > eui-2cf7f1c043500054

eui-2cf7f1c043500054

ID: eui-2cf7f1c043500054

↑ n/a ↓ n/a * Last activity 12 seconds ago

Overview Live data Messaging Location Payload formatters Claiming General settings

General information

End device ID	eui-2cf7f1c043500054
Frequency plan	Europe 863-870 MHz (SF9 for RX2 - recommended)
LoRaWAN version	LoRaWAN Specification 1.0.2
Regional Parameters version	RP001 Regional Parameters 1.0.2 revision B

Created at: Nov 7, 2022 19:50:34

Live data

↑ 19:50:46 Forward join-accept message DevAddr: 01 F1 E9 39

↓ 19:50:45 Accept join-request DevAddr: 01 F1 E9 39

See all activity →

Hardware

Brand	sensecap
Model	sensecaps2104-soil-moisture-temp
Hardware version	1.0
Firmware version	1.0

Activation information

AppEUI	2C F7 F1 C0 43 50 00 54
DevEUI	2C F7 F1 C0 43 50 00 54
AppKey	*****

Location

Change location settings →

No location information available

< Hide sidebar

On the gateway Live Data section you will see the data received by the gateway

The screenshot shows the The Things Stack dashboard interface. The top navigation bar includes tabs for 'Overview', 'Applications', 'Gateways' (which is selected), and 'Organizations'. A sidebar on the left lists 'ARM DEV SUMMIT' sections: Overview, Live data (selected), Location, Collaborators, API keys, and General settings. The main content area displays 'Live data' for the 'ARM DEV SUMMIT' gateway. It shows a table with columns for 'Time', 'Type', and 'Data preview'. Five uplink messages are listed:

Time	Type	Data preview
↑ 01:10:58	Receive uplink message	JoinEUI: 80 00 00 00 00 00 09 DevEUI: 2C F7 F1 C0 43 50 03 D9 Data rate: SF12BW125 SNR: 7 RSSI: -31
↑ 01:07:58	Receive uplink message	JoinEUI: 80 00 00 00 00 00 09 DevEUI: 2C F7 F1 C0 43 50 03 D9 Data rate: SF12BW125 SNR: 7 RSSI: -26
↑ 01:02:49	Receive uplink message	DevAddr: 48 00 08 A5 FCnt: 9 FPort: 2 Data rate: SF12BW125 SNR: 11.75 RSSI: -34
↑ 01:02:40	Receive uplink message	DevAddr: 48 00 08 A5 FCnt: 8 FPort: 2 Confirmed uplink Data rate: SF12BW125 SNR: 11 RSSI: -33
↑ 01:02:31	Receive uplink message	DevAddr: 48 00 08 A5 FCnt: 7 FPort: 2 Confirmed uplink Data rate: SF12BW125 SNR: 7 RSSI: -34

At the bottom of the main content area, there are buttons for 'Verbose stream' (with a toggle switch), 'Export as JSON', 'Pause', and 'Clear'. The footer of the dashboard includes links for 'Hide sidebar', 'EN', 'v3.21.0', and 'Documentation'.

On the application section you will see the decoded data received at the LNS by the LoRa node through the gateway

arm
DEVSUMMIT

The screenshot shows the The Things Stack application dashboard. The top navigation bar includes tabs for 'Overview', 'Applications' (which is selected), 'Gateways', and 'Organizations'. A sidebar on the left lists various categories: Overview, End devices, Live data, Payload formatters, Uplink, Downlink, Integrations, MQTT, Webhooks, Pub/Subs, LoRa Cloud, Collaborators, API keys, General settings, and a 'Hide sidebar' option. The main content area displays a table of 'Live data' for a 'Temperature' entity. The columns are 'Time', 'Entity ID', 'Type', 'Data preview', and 'Verbose stream' (with options to 'Export as JSON', 'Pause', and 'Clear'). The table lists numerous uplink data messages from a device with Entity ID 'eui-2cf7f1c043500054'. Each message row includes a DevAddr (01 95 AB AB) and a detailed payload description. The data preview shows a JSON object with 'err: 0', 'messages: []', and a long hex string for the payload. The verbose stream shows the raw hex data. The timestamp for the first message is 15:21:06.

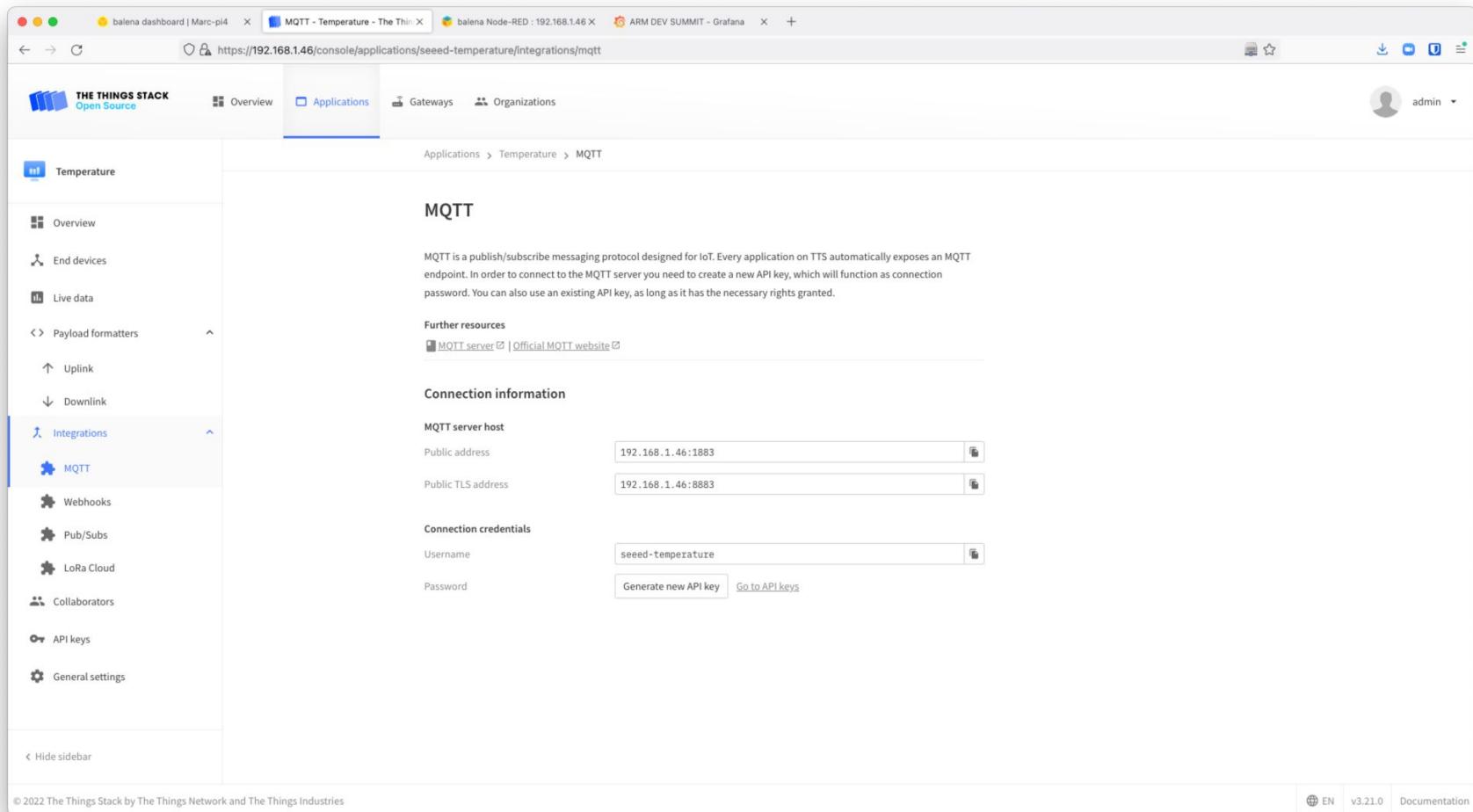
Applications > Temperature > Live data				
	Time	Entity ID	Type	Data preview
Overview	↑ 15:21:06	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "010610E4570000010710C4220000331A", valid: true } 01 06 10 E4 57 00 00 01
End devices	↑ 15:11:04	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "010610B8570000010710B842D0000086B7B", valid: true } 01 06 10 B8 57 00 00 01
Live data	↑ 15:01:06	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "010610B8570000010710182E000006A1", valid: true } 01 06 10 B8 57 00 00 01
Payload formatters	↑ 14:51:06	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "0106101C570000010710742700006932", valid: true } 01 06 10 1C 57 00 00 01
Uplink	↑ 14:41:06	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "0106101C570000010710B88240000E071", valid: true } 01 06 10 1C 57 00 00 01
Downlink	↑ 14:30:58	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "010610B8560000010710730000F648", valid: true } 01 06 10 B8 56 00 00 01
Integrations	↑ 14:21:04	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "010610B85600000107107100008786", valid: true } 01 06 10 B8 56 00 00 01
MQTT	↑ 14:11:01	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "00070064000A000106101C570000010710442F0000832C", valid: true } 00 07 00 00 01
Webhooks	↑ 14:00:59	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "0106101C570000010710C02B0000BA84", valid: true } 01 06 10 1C 57 00 00 01
Pub/Subs	↑ 13:51:01	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "0106101C5700000107105424000005E2", valid: true } 01 06 10 1C 57 00 00 01
LoRa Cloud	↑ 13:40:59	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "0106101C570000010710B002B0000B0E", valid: true } 01 06 10 1C 57 00 00 01
Collaborators	↑ 13:31:09	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "0106101C5700000107108841C00004AE", valid: true } 01 06 10 1C 57 00 00 01
API keys	↑ 13:21:03	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "0106101C57000001071068100000576C", valid: true } 01 06 10 1C 57 00 00 01
General settings	↑ 13:11:09	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "0106101C570000010710041000008762", valid: true } 01 06 10 1C 57 00 00 01
	↑ 13:00:59	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "0106101C570000010710CC10000086BC", valid: true } 01 06 10 1C 57 00 00 01
	↑ 12:51:07	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "010610B8560000010710AC0D00000002", valid: true } 01 06 10 B8 56 00 00 01
	↑ 12:41:09	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "010610F055000001071008087000000FB", valid: true } 01 06 10 F0 55 00 00 01
Hide sidebar	↑ 12:31:00	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "010610F0550000010710710C2E00008AAC", valid: true } 01 06 10 F0 55 00 00 01
	↑ 12:21:04	eui-2cf7f1c043500054	Forward uplink data message	DevAddr: 01 95 AB AB <> ↗ Payload: { err: 0, messages: [], payload: "010610F0550000010710710C2E00008AAC", valid: true } 01 06 10 F0 55 00 00 01

What we are going to build today?

- + Build a LoRaWAN Basics™ Station gateway with the Raspberry Pi, Seeed WM1302 concentrator, balena and The Things Stack LNS. And connect a LoRa node to The Things Stack LNS.
- + Build your own private LoRaWAN with The Things Stack taking the advantage of the edge with MING.



To send the data to NodeRED, go to Integrations and click on MQTT. Copy Username and Password



The screenshot shows the MQTT configuration page within the The Things Stack application. The left sidebar lists various integration options, with 'MQTT' currently selected. The main content area displays the MQTT connection setup, including fields for Public address (192.168.1.46:1883), Public TLS address (192.168.1.46:8883), Username (seed-temperature), and a button to Generate new API key.

THE THINGS STACK
Open Source

Overview Applications Gateways Organizations

admin

Temperature

Overview End devices Live data

Payload formatters

Uplink Downlink

Integrations

MQTT

Webhooks

Pub/Subs

LoRa Cloud

Collaborators

API keys

General settings

MQTT - Temperature - The Thing

balena Node-RED : 192.168.1.46 X

ARM DEV SUMMIT - Grafana X

https://192.168.1.46/console/applications/seed-temperature/integrations/mqtt

Applications > Temperature > MQTT

MQTT

MQTT is a publish/subscribe messaging protocol designed for IoT. Every application on TTS automatically exposes an MQTT endpoint. In order to connect to 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.

Further resources

[MQTT server](#) | [Official MQTT website](#)

Connection information

MQTT server host

Public address: 192.168.1.46:1883

Public TLS address: 192.168.1.46:8883

Connection credentials

Username: seed-temperature

Password: [Go to API keys](#)

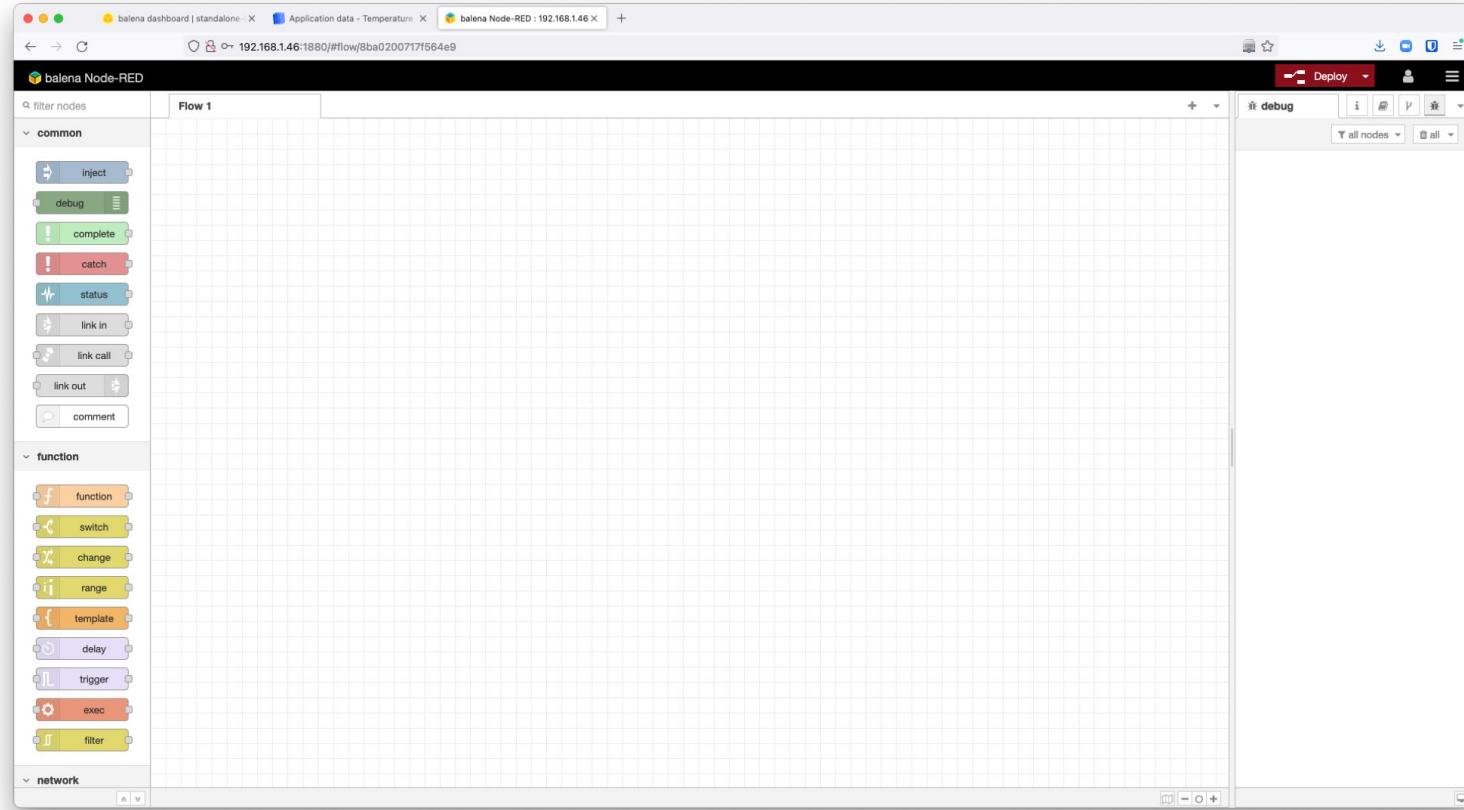
< Hide sidebar

© 2022 The Things Stack by The Things Network and The Things Industries

EN v3.21.0 Documentation

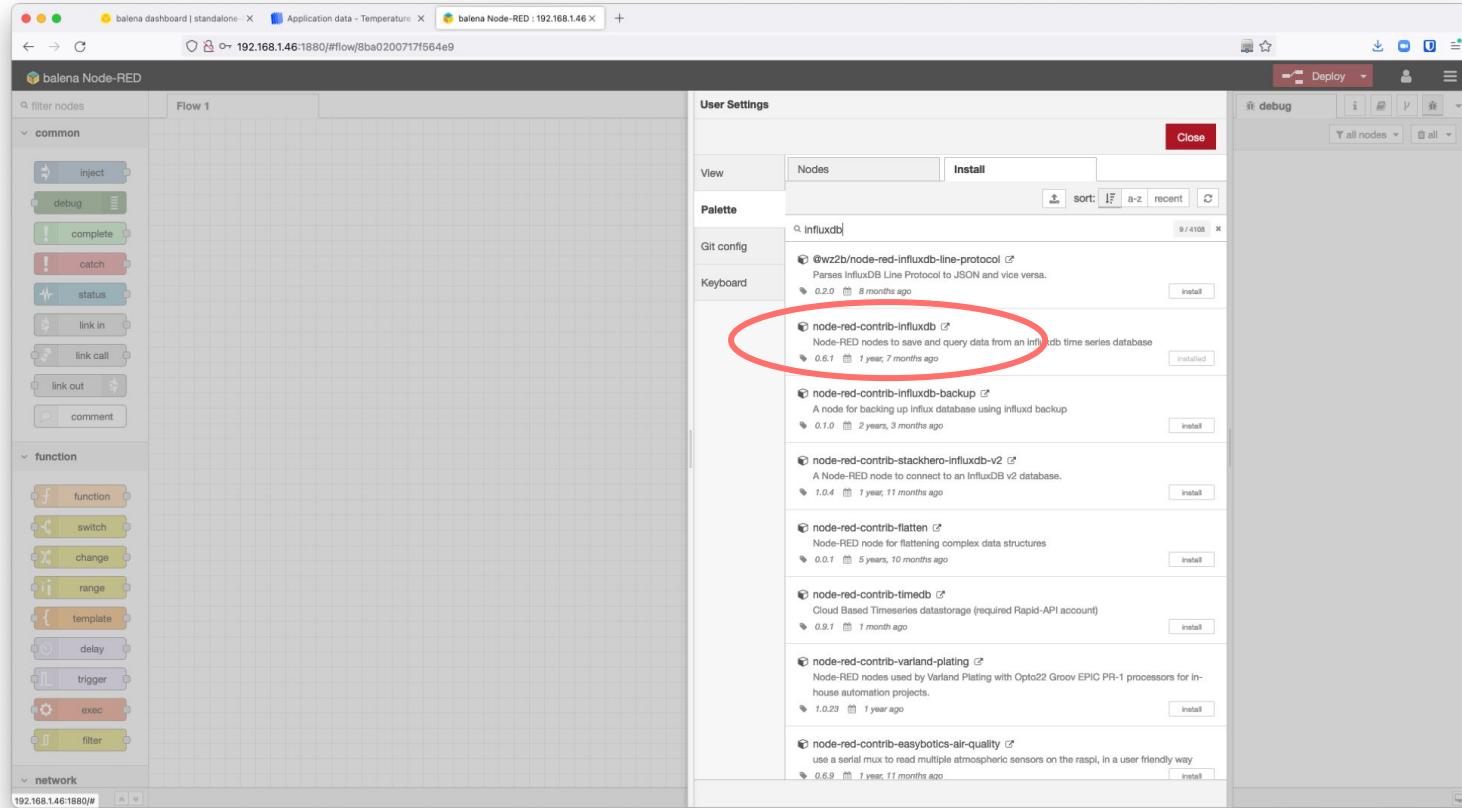
Go to <http://<your local ip address>:1880> and log in using balena / balena as username and password

arm
DEVSUMMIT



Install influxDB going to the menu “Manage palette” and going to the Install tab

arm
DEVSUMMIT



Add an MQTT node and connect to the “stack” service on Connection and go to Security

arm
DEVSUMMIT

The screenshot shows the balena Node-RED interface with a flow titled "Flow 1". The flow consists of the following nodes:

- An "inject" node connected to a MQTT node.
- The MQTT node has the topic "v3/+devices/+up" and is labeled "connected".
- The MQTT node connects to a "function" node.
- The "function" node has a single output connection.
- The output of the "function" node connects to a "msg" node.
- The "msg" node has an output connection labeled "[v1.x]".

The left sidebar shows categories for "common", "function", and "network" nodes. The "function" category includes nodes like "function", "switch", "change", "range", "template", "delay", "trigger", "exec", and "filter".

The right side of the interface displays the "Edit mqtt in node" configuration dialog for the MQTT node. The "Properties" section includes:

- Server: stack:1883
- Action: Subscribe to single topic
- Topic: v3/+devices/+up
- QoS: 2
- Output: a parsed JSON object
- Name: Name

Below the dialog, there is a "debug" panel showing log entries:

```
8/11/2022 1:41:40 node: 5cbbc38ce3d4b76
v3/seeded-temperature/devices/eui-2cf7f1c043500054/up:
msg.payload : Object
> { temperature: 21.9, soil: 0 }

8/11/2022 1:51:34 node: 5cbbc38ce3d4b76
v3/seeded-temperature/devices/eui-2cf7f1c043500054/up:
msg : Object
> { topic: "v3/seeded-temperature
/devices/eui-2cf7f1c043500054", payload: object, qos: 0, retain: false, _msgid: "26672b9843812765" }

8/11/2022 2:01:38 node: 5cbbc38ce3d4b76
v3/seeded-temperature/devices/eui-2cf7f1c043500054/up:
msg : Object
> object
topic: "v3/seeded-temperature
/devices/eui-2cf7f1c043500054/up"
> payload: object
temperature: 22
soil: 0
qos: 0
retain: false
msgid: "2e6ec623693d2303"
measurement:
"eui-2cf7f1c043500054"

8/11/2022 2:11:30 node: 5cbbc38ce3d4b76
v3/seeded-temperature/devices/eui-2cf7f1c043500054/up:
msg : Object
> object
topic: "v3/seeded-temperature
/devices/eui-2cf7f1c043500054/up"
> payload: object
temperature: 22.2
soil: 0
qos: 0
retain: false
msgid: "cdaef6697ca0b57"
measurement:
"eui-2cf7f1c043500054"
```

At the bottom of the dialog, there is an "Enabled" checkbox.

Paste the credentials copied from The Things Stack MQTT integration

The screenshot shows the balena Node-RED interface running on a device with IP 192.168.1.46. The main area displays a flow titled "Flow 1" with a single node: a "function" node with the ID "v3/+devices/+up". This node has two outputs connected to a "debug" node. The "Properties" tab of the configuration panel for this node is open, showing the following settings:

- Name:** Name
- Connection:** seed-temperature
- Security:** None
- Messages:** None
- Username:** seed-temperature
- Password:** (redacted)

The "Messages" tab is collapsed. On the right side of the screen, the "debug" tab is active, displaying log entries from the Node-RED instance. The logs show MQTT messages being published to the topic "v3/seeed-temperature/devices/eul-2cf7f1c043500054/up". The first message is at 8/1/2022 1:41:40 and the second is at 8/1/2022 1:51:34. Both messages contain a payload object with temperature and soil properties.

```
8/1/2022 1:41:40 node: 5cbcb38ce3d4b76
v3/seeed-temperature/devices/eul-2cf7f1c043500054/up:
msg.payload : Object
> { temperature: 21.9, soil: 0 }

8/1/2022 1:51:34 node: 5cbcb38ce3d4b76
v3/seeed-temperature/devices/eul-2cf7f1c043500054/up:
msg : Object
> { topic: "v3/seeed-temperature
/devices/eul-2cf7f1c043500054", payload: object, qos: 0, retain: false, _msgid: "26672b9843b12765" }

8/1/2022 2:01:38 node: 5cbcb38ce3d4b76
v3/seeed-temperature/devices/eul-2cf7f1c043500054/up:
msg : Object
> object
topic: "v3/seeed-temperature
/devices/eul-2cf7f1c043500054"
payload: object
temperature: 22
soil: 0
qos: 0
retain: false
msgid: "2e6ec623693d2303"
measurement:
"eul-2cf7f1c043500054"

8/1/2022 2:11:30 node: 5cbcb38ce3d4b76
v3/seeed-temperature/devices/eul-2cf7f1c043500054/up:
msg : Object
> object
topic: "v3/seeed-temperature
/devices/eul-2cf7f1c043500054"
payload: object
temperature: 22.2
soil: 0
qos: 0
retain: false
msgid: "cdaef6697ca@b57"
measurement:
"eul-2cf7f1c043500054"
```

Add an InfluxDB out node and connect to the “influxdb” host service running on your device. The database is “balena”

arm
DEVSUMMIT

The screenshot shows the balena Node-RED interface with a flow titled "Flow 1". The flow consists of the following nodes:

- An "inject" node connected to a "v3/+devices/+up" MQTT node.
- A "function" node connected between the MQTT node and a "msg" node.
- The "msg" node is connected to an "InfluxDB out" node.

The "InfluxDB out" node configuration is as follows:

- Name: Name
- Version: 1.x
- Host: influxdb
- Port: 8086
- Database: balena
- Username: (empty)
- Password: (empty)
- Enable secure (SSL/TLS) connection:

On the right side of the interface, there is a "debug" panel displaying log entries:

```
8/11/2022 1:41:40 node: 5cfcfb38ce03d4b76 v3/seeded-temperature/devices/eul-2cf7f1c043500054/up: msg.payload : Object { temperature: 21.9, soil: 0 } 8/11/2022 1:51:34 node: 5cfcfb38ce03d4b76 v3/seeded-temperature/devices/eul-2cf7f1c043500054/up: msg : Object { topic: "v3/seeded-temperature /devices/eul-2cf7f1c043500054", payload: object, qos: 0, retain: false, _msgid: "26672b9843b12765" } 8/11/2022 2:01:38 node: 5cfcfb38ce03d4b76 v3/seeded-temperature/devices/eul-2cf7f1c043500054/up: msg : Object { topic: "v3/seeded-temperature /devices/eul-2cf7f1c043500054", payload: object, temperature: 22, soil: 0, qos: 0, retain: false, _msgid: "2e6ec623693d2303" } measurement: "eul-2cf7f1c043500054" 8/11/2022 2:11:30 node: 5cfcfb38ce03d4b76 v3/seeded-temperature/devices/eul-2cf7f1c043500054/up: msg : Object { topic: "v3/seeded-temperature /devices/eul-2cf7f1c043500054", payload: object, temperature: 22.2, soil: 0, qos: 0, retain: false, _msgid: "c6daef6697ca0b57" } measurement: "eul-2cf7f1c043500054"
```

Go to Terminal and connect to the “influxdb” service. Get into the “influx” CLI and type “create database balena”

arm
DEVSUMMIT

The screenshot shows the balenaCloud dashboard interface for a device named "Marc-pi4".

Device Summary:

- Status:** Online (UUID: b9fbcd4)
- OS:** Raspberry Pi 4 (using 64bit OS)
- Online For:** about 14 hours
- Supervisor Version:** 14.2.0
- Support Access:** Off
- Local IP Address:** 192.168.1.46
- Public IP Address:** 79.153.119.0
- Tags:** EUI:DCA632FFF554B34, URL: https://192.168.1.46
- Notes:** EUI : DCA632FFF554B34

Services:

Service	Status	Release
basestation	Running	d3542ad
grafana	Running	d3542ad
influxdb	Running	d3542ad
node-red	Running	d3542ad
postgres	Running	d3542ad
redis	Running	d3542ad
stack	Running	d3542ad
wifi-connect	Running	d3542ad

Metrics:

- CPU: ~76% Usage
- Temperature: ~60C
- Memory: 1.1 GB / 3.8 GB
- Storage: 2.7 GB / 28.3 GB

Logs:

```
08.11.22 01:17:02 (+0000) influxdb [2022-11-08T01:17:02.409762Z] info Executing query ("log_id": "0e0rS0IG 000", "service": "query", "query": "SELECT fieldKey, fieldType FROM balena.autogen._fieldKeys")  
08.11.22 01:17:02 (+0000) influxdb [httpd] 172.18.0.5 - - [08/Nov/2022:01:17:02 +0000] "GET /query?db=balena&q=sho wField&keys HTTP/1.1" 200 278 "-" "Python-urllib/3.7" 8ce2ef34-5f03-11ed-8986-0242ac120004 3629  
08.11.22 01:17:05 (+0000) grafana [Interim dashboard sync complete.  
08.11.22 01:17:05 (+0000) basestation 2022-11-08 01:17:05.155 [SYN:VERB] Time sync rejected: quality=286 threshold=267  
08.11.22 01:17:12 (+0000) influxdb [2022-11-08T01:17:12.797653Z] info Executing query ("log_id": "0e0rS0IG 000", "service": "query", "query": "SELECT fieldKey, fieldType FROM balena.autogen._fieldKeys")  
08.11.22 01:17:12 (+0000) influxdb [httpd] 172.18.0.5 - - [08/Nov/2022:01:17:12 +0000] "GET /query?db=balena&q=sho wField&keys HTTP/1.1" 200 278 "-" "Python-urllib/3.7" 131401bf-5f03-11ed-8987-0242ac120004 1727  
08.11.22 01:17:13 (+0000) grafana [Interim dashboard sync complete.  
08.11.22 01:17:15 (+0000) basestation 2022-11-08 01:17:15.658 [SYN:VERB] Time sync rejected: quality=274 threshold=267
```

Terminal:

```
Connecting to b9fbcd42b0de09e5b65aa77b2edc1ff...  
Spawning shell...  
root@27005dcbb608:/# influx  
Connected to http://localhost:8086 version 1.7.11  
InfluxDB shell version: 1.7.11  
> create database balena  
> 
```

Edit the function node

The screenshot shows the balena Node-RED interface running on a device at 192.168.1.46:1880. The left sidebar contains categories for common nodes (inject, debug, complete, catch, status, link in, link call, link out, comment) and function nodes (function, switch, change, range, template, delay, trigger, exec, filter). The main canvas displays a flow titled "Flow 1". The flow starts with a "v3/+devices/+/up" node connected to a "function" node. The output of the function node goes to an "msg" node, which then connects to an "[v1.x] influxdb:8086/balena" node. The right side of the interface features a "debug" panel showing log entries from the Node-RED instance:

```
8/1/2022 1:41:40 node: 5cfcfb38ce3d4b76 v3/seeded-temperature/devices/eui-2cf7f1c043500054/up: msg.payload : Object > { temperature: 21.9, soil: 0 } 8/1/2022 1:51:34 node: 5cfcfb38ce3d4b76 v3/seeded-temperature/devices/eui-2cf7f1c043500054/up: msg : Object > { topic: "v3/seeded-temperature /devices/eui-2cf7f1c043500054", payload: object, qos: 0, retain: false, _msgid: "26672b9843b12765" } 8/1/2022 2:01:38 node: 5cfcfb38ce3d4b76 v3/seeded-temperature/devices/eui-2cf7f1c043500054/up: msg : Object > object topic: "v3/seeded-temperature /devices/eui-2cf7f1c043500054/up" > payload: object temperature: 22 soil: 0 qos: 0 retain: false _msgid: "2e6ec623693d2303" measurement: "eui-2cf7f1c043500054"
```

This is a hack for the SenseCAP S2104 LoRa node to get the data to InfluxDB and visualized easier on Grafana

arm
DEVSUMMIT

The screenshot shows the balena Node-RED interface with a flow titled "Flow 1". The flow consists of a "v3/+devices/+up" node connected to a "function" node. The "function" node has its code editor open, displaying the following JavaScript code:

```
1 var decoded = {};
2
3 var message = msg.payload.uplink_message.decoded_payload.messages;
4 var device = msg.payload.end_device_ids.device_id;
5
6 var temperature = 0;
7 var soil = 0;
8
9 for (var i = 0; i < message.length; i++) {
10    var item = message[i].measurementId;
11    if (item == 4102) {
12    {
13        temperature = message[i].measurementValue;
14    }
15    else if (item == 4103) {
16        {
17            soil = message[i].measurementValue;
18        }
19    }
20}
21 msg.measurement = device;
22 msg.payload = {"temperature": temperature, "soil": soil};
23
24 return msg;
```

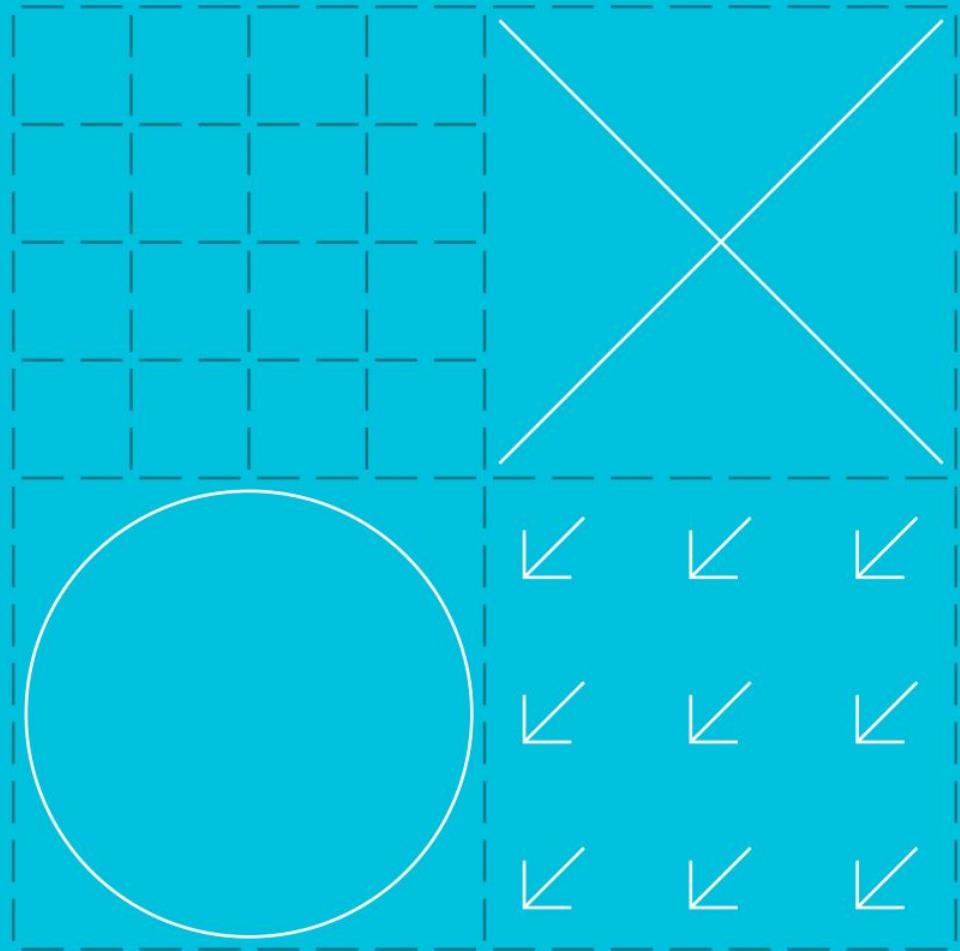
The right side of the interface shows a "debug" panel with log entries from the Node-RED instance running on 192.168.1.46. The logs include messages received from a device with EUI-64 "eui-2cf7f1c04350054" at various times, containing temperature and soil moisture data.

Go to <http://<your local ip address>:3000> and sign in using admin / admin credentials. InfluxDB appears as a default datasource. Feel free to visualize the data.

arm
DEVSUMMIT



Summary



- LPWAN are here ready to use!
- Reduce the friction to manage your fleet, the easy way™.
- Use modern open source tools to get benefit of the edge computing.

- LPWAN are here ready to use!
- Reduce the friction to manage your fleet, the easy way™.
- Use modern open source tools to get benefit of the edge computing.
- **Please, contribute!**



Marc Pous

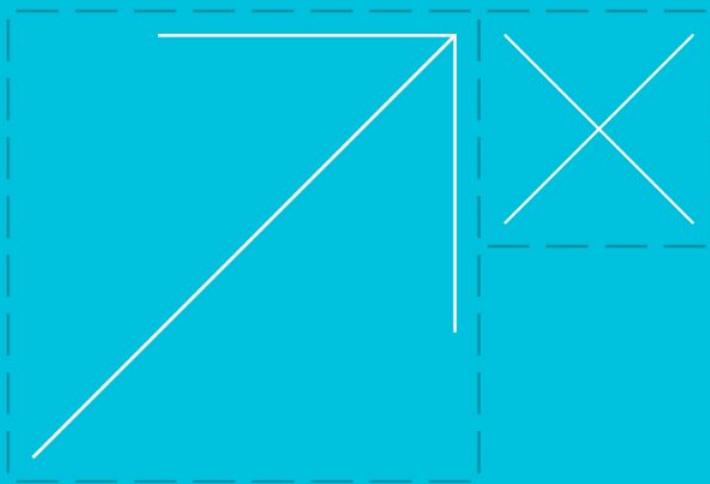
Developer Advocate @ **balena.io**
Initiator **IoT Barcelona** and **IoT Stars**

e: marc@balena.io
t: [@gy4nt](https://twitter.com/@gy4nt)
g: github.com/mpous

Questions?

and feedback?

<https://forums.balena.io/>



Thank You

Gràcies

Danke

Gracias

謝謝

ありがとう

Asante

Merci

감사합니다

ধন্যবাদ

Kiitos

شکرًا

ধন্যবাদ

האודה

**Learn what is LoRa,
LoRaWAN and build your
first LoRaWAN gateway,
the easy way™**

Marc Pous - Developer Advocate at balena.io

arm Dev Summit 2022



The Arm trademarks featured in this presentation are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

www.arm.com/company/policies/trademarks

