

# Building a Meshtastic Internet Gateway

A Meshtastic internet gateway bridges local LoRa radio traffic to the internet and can serve as a powerful community infrastructure node. This guide covers setting up a dedicated gateway on a Raspberry Pi.

## Gateway hardware options

Option	Hardware	Pros	Cons
<b>ESP32 node (simplest)</b>	Heltec V3/V4, T-Beam	Single device, no Pi needed, compact	Single-channel, limited processing
<b>Pi + LoRa hat</b>	Raspberry Pi 4 + SX126x HAT	Full Linux environment via meshtasticd	More complex setup. Note: RAK2287/RAK5146 are LoRaWAN concentrators; Meshtastic on Linux (meshtasticd) uses a single-radio SX126x HAT, not a multi-channel LoRaWAN concentrator.
<b>Pi + USB LoRa node</b>	Raspberry Pi + RAK4631 USB	Easy setup, use standard Meshtastic firmware	Single channel only

## Option 1: Dedicated ESP32 gateway node

The simplest approach: configure a Heltec V3/V4 or T-Beam as a dedicated gateway. This node doesn't need to be a router/repeater - its primary job is bridging radio and MQTT.

1. Flash with Meshtastic firmware (standard)
2. Connect to your home or community WiFi: `meshtastic --set network.wifi_enabled true --set network.wifi_ssid "YourSSID" --set network.wifi_psk "YourPassword"`
3. Configure MQTT as described in the MQTT Setup page

4. Leave the role at the default `CLIENT`. Do **not** set this gateway to ROUTER: on ESP32, ROUTER force-enables power-saving sleep and defaults WiFi/BLE/Serial OFF, which breaks a WiFi MQTT gateway. ROUTER is only for well-sited dedicated infrastructure radios, not for a node whose job is bridging to MQTT.
5. Mount at a good location with LoRa antenna and reliable WiFi

**Privacy warning:** A gateway that uplinks the default LongFast channel publishes *all* traffic it hears - including node positions - to the public internet, and (despite the channel PSK) Meshtastic uploads to MQTT **unencrypted by default** unless you set `mqtt.encryption_enabled true`. Use a private channel with encryption enabled, or accept that default-channel traffic is fully public. Set `lora.ignore_mqtt true` to prevent rebroadcast loops.

## Option 2: Raspberry Pi + USB LoRa node

Connect an nRF52840-based device (RAK4631, T-Echo) via USB to a Raspberry Pi. The Pi handles internet connectivity while the LoRa node handles radio.

### Setup the LoRa node

```
# Flash with Meshtastic firmware
# Leave the role at the default CLIENT for a USB-attached gateway radio.
# Do NOT set ROUTER, which disables serial and forces sleep on ESP32.
```

## Install meshtasticd (Meshtastic daemon)

The Python CLI installs from PyPI, but `meshtasticd` is **not** on PyPI - install it from the official Meshtastic apt repository:

```
# Python CLI (PyPI)
pip3 install meshtastic

# meshtasticd daemon (apt repo, not pip)
sudo add-apt-repository ppa:meshtastic/beta
sudo apt update
sudo apt install meshtasticd
```

# Configure MQTT bridging via Python

```
import meshtastic
import meshtastic.serial_interface
from pubsub import pub
import paho.mqtt.client as mqtt
import json

# Connect to the LoRa node
iface = meshtastic.serial_interface.SerialInterface("/dev/ttyUSB0")

# Connect to MQTT broker
mq = mqtt.Client()
mq.username_pw_set("meshdev", "large4cats")
mq.tls_set()
mq.connect("mqtt.meshtastic.org", 8883)

# Forward all received packets to MQTT.
# Topic format: msh/REGION/2/json/CHANNELNAME/USERID
# (version "2" comes before the channel name; packet type is a field inside the JSON, not a
topic segment)
def on_receive(packet, interface):
    userid = packet.get('fromId', '?')
    topic = f"msh/US/2/json/LongFast/{userid}"
    mq.publish(topic, json.dumps(packet))

# Packets are delivered via PyPubSub, not an iface.on_receive attribute
pub.subscribe(on_receive, "meshtastic.receive")

mq.loop_forever()
```

**Note:** `meshdev` / `large4cats` is the shared *public* broker, which is rate-limited, filtered, and not an open firehose. The firmware's built-in MQTT uplink is the supported, recommended path; this raw-publish script is illustrative only.

## Monitoring your gateway

A healthy gateway should be publishing to MQTT continuously. Monitor with:

```
# Subscribe to your node's traffic (replace !abcd1234 with your node ID)
mosquitto_sub -h mqtt.meshtastic.org -p 8883 -t "msh/US/#" -u meshdev -P large4cats --tls-use-os-certs | grep abcd1234
```

You should see JSON packets appearing whenever your node hears a packet on the mesh. If the stream is silent for more than a few minutes in an active network, your WiFi or MQTT connection may have dropped.

# Adding your gateway to the community map

To appear on third-party maps, your gateway needs MQTT uplink working plus map reporting enabled in the MQTT module config (`mapReportingEnabled`, firmware  $\geq 2.3.2$ ) and a fixed or GPS position set. Map reports publish at most about hourly, so allow up to an hour or more before your node shows up. Appearance on [meshmap.net](https://meshmap.net) depends on that third-party site and is not instantaneous or guaranteed; verify by searching for your node name.

---

Revision #3

Created 2026-05-03 03:54:16 UTC by Mesh America Admin

Updated 2026-06-09 12:13:52 UTC by Mesh America Admin