Rooftop Gateway Build (Pi + LoRa)

A rooftop gateway bridges your local LoRa mesh to the internet, enabling remote monitoring via meshmap.net, MQTT integration with Home Assistant, and APRS forwarding. This build uses a Raspberry Pi Zero 2W paired with a USB-connected LoRa node as the simplest, most maintainable approach.

⚠ **ROOFTOP WORK SAFETY - READ FIRST:** Rooftop installation carries serious fall and electrical risk.

- **Fall protection:** Use a full-body harness with a fall-arrest anchor when working within 6 ft of any roof edge or skylight. Never work on wet, icy, steep, or fragile roofs.
- **Power lines:** Survey for overhead service-drop power lines before raising any mast or antenna. Keep the antenna/mast clear of all lines by at least its full height plus a margin (the mast's full fall-radius) - contact with a power line is the leading cause of installer fatalities.
- **Never work alone:** Have a second person present.
- **Grounding:** Bond the antenna ground rod to the building grounding electrode system (per NEC 810.21 / 250).
- If in doubt, hire a qualified installer.

Parts List

Part	Approx. Cost
Raspberry Pi Zero 2W	~$15
Heltec LoRa 32 V3 (SX1262, USB-C) - acts as the LoRa radio	~$15 - $25 (street price varies; as of 2026-06-08)
LoRa antenna (915 MHz, SMA, matched to the Heltec) + pigtail	~$8 - $15
5V PoE splitter (802.3af to micro-USB/USB-C) *or* USB power supply	~$10
MicroSD card, 16 GB (Class 10 / A1 or better)	~$8

Part	Approx. Cost
Weatherproof outdoor enclosure (IP65 or better, fits Pi + Heltec) - light-colored / shaded	~$25
Short USB-A to USB-C cable (internal, ~15 cm)	~$3
Total	**~$84 - 115 (estimate, subject to current street pricing)**

Connect the antenna to the Heltec before powering it on (good practice for any LoRa radio). **Thermal note:** a sealed enclosure in direct rooftop sun can reach 70-80 °C internally - well above a Raspberry Pi's reliable operating range and hard on SD-card lifespan. Use a light-colored or white enclosure, mount it in shade where possible, and add rain-protected ventilation. See the Thermal Management for Outdoor Enclosures page for details.

Alternative radio option: For LoRaWAN instead of Meshtastic, substitute the Heltec with a RAK2287 Pi HAT (SX1302 8-channel concentrator, ~$80 as of 2026-06-08) and use the ChirpStack network server. This guide focuses on the Meshtastic MQTT gateway path.

Setup: Meshtastic MQTT Gateway

1. Prepare the Pi

Flash Raspberry Pi OS Lite (64-bit) to the microSD card using Raspberry Pi Imager. In the Imager advanced settings, pre-configure your Wi-Fi credentials, enable SSH, and set a hostname (e.g. `mesh-gateway`). This avoids needing a display or keyboard on first boot.

2. Connect the Heltec

Connect the Heltec LoRa 32 V3 to the Pi Zero 2W via the short USB-C cable. The Pi will enumerate the Heltec as a USB serial device. The Heltec V3's USB-serial bridge usually appears as `/dev/ttyUSB0` (native-USB nRF boards appear as `/dev/ttyACM0`). Confirm which one you actually have with:

```
ls /dev/tty{USB,ACM}*
```

Important: In every command below, substitute the port you actually found here (shown as `/dev/ttyXXX`). If your Heltec appeared as `/dev/ttyUSB0`, use that instead of `/dev/ttyACM0`, or the commands will silently fail against the wrong port.

3. Install Software

```
sudo apt update && sudo apt upgrade -y
pip install meshtastic
sudo apt install -y mosquitto mosquitto-clients
```

4. Configure the Heltec via Meshtastic CLI

Connect to the node over USB serial and enable MQTT. Note that `uplink_enabled` and `downlink_enabled` are **per-channel** settings (use `--ch-index` / `--ch-set`), not fields on the `mqtt` module:

```
# Replace /dev/ttyXXX with the port you found in step 2
# Set MQTT server to localhost (the Pi itself)
meshtastic --port /dev/ttyXXX --set mqtt.address localhost
meshtastic --port /dev/ttyXXX --set mqtt.enabled true
# Uplink/downlink are PER-CHANNEL - set them on the primary channel (index 0)
meshtastic --port /dev/ttyXXX --ch-index 0 --ch-set uplink_enabled true
meshtastic --port /dev/ttyXXX --ch-index 0 --ch-set downlink_enabled true
# Enable JSON output (optional, for Home Assistant compatibility)
meshtastic --port /dev/ttyXXX --set mqtt.json_enabled true
```

5. Configure Mosquitto

Edit `/etc/mosquitto/mosquitto.conf`. **Use authentication by default** - configure a username/password rather than an open anonymous listener:

```
listener 1883
allow_anonymous false
password_file /etc/mosquitto/passwd
```

Create the password file with `sudo mosquitto_passwd -c /etc/mosquitto/passwd meshuser`. An anonymous, unauthenticated broker (`allow_anonymous true`) is only acceptable on a fully trusted local network.

> ⚠ **Do NOT expose the MQTT broker to the public internet.** Open brokers are constantly scanned and abused. Never port-forward an unauthenticated broker. For remote access, use a VPN (e.g. WireGuard / Tailscale) into your network, or bridge to the community meshmap MQTT server - do not open port 1883 to the internet.

Restart Mosquitto:

```
sudo systemctl restart mosquitto
sudo systemctl enable mosquitto
```

6. Network Connectivity

Options in order of preference:

- **PoE Ethernet:** Use a PoE splitter to power the Pi over the same Ethernet cable that connects it to your router. Most reliable and simplest.
- **Wi-Fi:** The Pi Zero 2W has 2.4 GHz Wi-Fi. Works well if the rooftop is within range of your router. Add a second 2.4 GHz AP if needed.
- **Ethernet-over-USB (USB gadget mode):** Configure the Pi as a USB network adapter - plug a USB cable to a computer or router port. Useful when no other connectivity is available near the Pi.

7. Optional: Node-RED for Local Processing

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

Node-RED provides a visual flow editor for filtering, transforming, and routing mesh packets to Home Assistant, InfluxDB, or external webhooks without writing code.

8. Auto-Start on Boot (systemd)

The Meshtastic node's own firmware MQTT client publishes to the broker - this happens on the node itself, not via a service on the Pi. For packets to publish, you need **both** `mqtt.enabled = true` **and** at least one channel with `uplink_enabled = true` (set in step 4). When both are in place and Mosquitto is running, no custom systemd service is needed; but if you skipped the per-channel uplink step you will see no packets. If you add a custom Python script (e.g. for APRS forwarding), create a systemd service:

```
# /etc/systemd/system/mesh-bridge.service
[Unit]
Description=Mesh MQTT Bridge
After=network.target mosquitto.service
```

```
Requires=mosquitto.service

[Service]
ExecStart=/usr/bin/python3 /home/pi/mesh_bridge.py
Restart=always
RestartSec=10
User=pi

[Install]
WantedBy=multi-user.target
```

```
sudo systemctl enable mesh-bridge
sudo systemctl start mesh-bridge
```

9. Verify Packet Flow

Subscribe to all Meshtastic topics on the local broker and confirm packets are arriving:

```
mosquitto_sub -h localhost -t 'msh/#' -v
```

You should see JSON or binary payloads appearing whenever a node in range transmits. If nothing appears, check USB serial connectivity and that the channel uplink is enabled on the Heltec.

Use Cases

- **meshmap.net visibility:** Configure Mosquitto to bridge to the public meshmap MQTT server so your nodes appear on the community map. See the meshmap.net documentation for bridge configuration details.
- **Home Assistant integration:** Use the Mosquitto add-on in Home Assistant and subscribe to `msh/2/json/#` for parsed telemetry and position data. Create automations triggered by mesh events.
- **APRS gateway:** Run `aprx` or a custom script to re-encode position packets as APRS-IS frames and upload to aprs.fi for interoperability with the ham radio APRS network. Uploading to APRS-IS requires a valid amateur callsign and passcode. If you also gate this traffic onto APRS RF, Part 97 rules apply to the transmitted traffic - you must identify your station (97.119) and must not transmit encrypted content (97.113).
- **Remote node monitoring:** Query node telemetry via MQTT to check battery voltage, SNR, and uptime of your remote repeaters. Access this over a VPN into your network rather than exposing the broker to the internet (see the security warning in step 5).

Revision #3
Created 2026-05-03 05:28:11 UTC by Mesh America Admin
Updated 2026-06-08 23:34:59 UTC by Mesh America Admin