Setting Up Meshtastic

Step-by-step device configuration for Meshtastic.

- Meshtastic Setup Guide
- Initial Node Configuration Checklist
- Advanced Configuration for Infrastructure Nodes

Meshtastic Setup Guide

Getting started with Meshtastic is straightforward. The project has extensive documentation and a large community, making it one of the easiest LoRa platforms to get up and running.

What you need

- A Meshtastic-compatible LoRa device, fully charged
- A smartphone (Android or iOS) or a computer
- The **Meshtastic app** (free on Google Play and the App Store) or the web client at client.meshtastic.org

Step 1 - Flash firmware (if needed)

Many devices ship without Meshtastic pre-installed. Use the **Meshtastic Web Flasher** at flasher.meshtastic.org to install firmware directly from your browser - no special software required.

1. Connect your device via USB
2. Open flasher.meshtastic.org in Chrome or Edge
3. Select your device type and click Flash
4. Wait for the process to complete and reboot

Step 2 - Connect via app or web client

Open the Meshtastic app on your phone. The app will scan for nearby Bluetooth devices. Select your node, pair, and the app will show you the node configuration screen.

Alternatively, use the web client at client.meshtastic.org if you prefer a browser-based interface.

Step 3 - Configure region

This step is mandatory before the radio will transmit. In the app or web client:

1. Go to **Config → LoRa**
2. Set **Region** to the code that matches your country - **US** for the United States *and Canada* (both use the 902-928 MHz US915 band), **EU_868** for most of Europe, and so on. See the region reference for other countries.
3. Save and allow the device to reboot

Setting the wrong region transmits on the wrong frequencies, which may be illegal and will prevent you from connecting to other local nodes.

Step 4 - Set your node name

In **Config → Device**, set a recognizable long name and short name for your node. Other users on the network will see this name.

Step 5 - Test

Open the **Messages** tab and send a message on the default channel. Nearby Meshtastic nodes will receive it. You can also check the **Nodes** tab to see any other devices currently in range.

Note: the default channel is *public* - anyone running Meshtastic in range (or anyone watching the public MQTT broker, if a gateway uplinks it) can read it. It is fine for testing, but do not send anything sensitive on it. Create a private channel with its own key for private conversations.

Further configuration

Meshtastic has many configurable parameters covering radio settings, power management, display, telemetry, and more. The official Meshtastic documentation at meshtastic.org/docs is comprehensive and well-maintained - refer to it for anything beyond basic setup.

Initial Node Configuration Checklist

When you get a new Meshtastic node, running through a standard configuration checklist ensures it's properly set up for your network before deployment. This guide covers every setting that matters for a production deployment.

Step 1: Flash Latest Stable Firmware

Even brand-new hardware may ship with outdated firmware. Always flash the latest stable release before configuration:

1. Visit flasher.meshtastic.org in Chrome or Edge
2. Connect node via USB
3. Select your board model exactly (wrong board = failed flash)
4. Select "Latest Release" and click Flash
5. Wait for flash to complete (1-3 minutes)

Step 2: Region Setting (Critical)

The region setting controls which frequency band the node uses. Transmitting on the wrong frequency is illegal and will cause your node to interfere with other services. The region setting is the master legal cap on both frequency and transmit power: selecting a region other than your own (for example a foreign region preset) can put your node outside your country's authorized band and power limits.

```
# For US users:
meshtastic --set lora.region US

# For Canada:
meshtastic --set lora.region US  # Canada uses the same 902-928 MHz (US915) band as the US
```

```
# Check current setting:
meshtastic --get lora.region
```

The node will not transmit until a region is set. This is intentional - it prevents out-of-box interference.

Step 3: Set Your Identity

```
# Set your long name (visible to all nearby nodes):
meshtastic --set-owner "Your Name or Node Name"

# Set your short name (4 chars, shown on map):
meshtastic --set-owner-short "AB01"
```

Step 4: Configure Your Channel

To join an existing community network, the easiest and safest method is to scan their QR code (or open their shared channel URL) with the Meshtastic app, which imports the channel name and key automatically. If you configure the channel manually instead, you must use the community's *real* base64 PSK - the placeholder below is not valid base64 and typing it verbatim will error or set a garbage key:

```
# Set channel name and key (replace base64encodedkey== with your community's REAL base64 PSK):
meshtastic --ch-index 0 --ch-set name "CommunityName"
meshtastic --ch-index 0 --ch-set psk "base64encodedkey=="

# Alternatively, for a new/default key instead of a shared one:
#   --ch-set psk none     (no encryption)
#   --ch-set psk random   (generate a fresh random key)

# Verify channel settings:
meshtastic --info
```

Step 5: Set Device Role

```
# For a personal/portable node:
meshtastic --set device.role CLIENT

# For a fixed home station that also relays traffic, use CLIENT (smart relay) or ROUTER_LATE.
# ROUTER_CLIENT is deprecated and removed in current firmware - do not use it:
meshtastic --set device.role CLIENT

# For a dedicated infrastructure repeater:
meshtastic --set device.role ROUTER
```

Step 6: Configure GPS (if equipped)

```
# Enable GPS position broadcasting (current firmware uses the gps_mode enum;
# older firmware used the boolean position.gps_enabled):
meshtastic --set position.gps_mode ENABLED

# Set position broadcast interval (seconds):
# 900 = every 15 minutes (good for fixed nodes)
# 60 = every minute (for mobile use)
meshtastic --set position.position_broadcast_secs 900

# For fixed installations, set a fixed position instead of using GPS:
meshtastic --setlat 45.5051 --setlon -122.6750 --setalt 50
```

Step 7: Verify and Test

```
# Show full device info:
meshtastic --info

# Listen for incoming packets:
meshtastic --listen

# Send a test message:
```

```
meshtastic --sendtext "Test from [your name]"
```

Check that your node appears in the app's node list and that the channel matches your community's configuration. Send a test message and verify it's received by another node.

Advanced Configuration for Infrastructure Nodes

Infrastructure nodes (routers, backbone repeaters) require additional configuration beyond the defaults to operate efficiently and reliably in a production network.

Power Management

The ROUTER role automatically uses power-saving sleep on ESP32 boards (the LoRa radio stays in standby so it can wake on incoming packets) - this cannot be disabled. If you need BLE, WiFi, or serial to stay continuously available on an infrastructure node, use the CLIENT or ROUTER_LATE role instead. The settings below affect non-router nodes:

```
# Light-sleep tuning (note: power.is_power_saving false is already the
# default for non-router roles, but ROUTER force-enables sleep on ESP32):
meshtastic --set power.is_power_saving false
meshtastic --set power.ls_secs 0 # 0 selects the DEFAULT 5-minute light-sleep interval (it
does NOT disable light sleep)
```

Telemetry Configuration

Infrastructure nodes can report their own health metrics to the network:

```
# Enable device telemetry (battery, voltage, uptime) - firmware default is 1800 s (30 min):
meshtastic --set telemetry.device_update_interval 1800 # every 30 min

# Enable environment telemetry (if sensor equipped):
meshtastic --set telemetry.environment_update_interval 1800

# Reduce position airtime on a fixed node by setting a long broadcast interval
# (setting 0 does NOT disable broadcasts):
meshtastic --set position.position_broadcast_secs 900
```

Hop Limit for Infrastructure Nodes

Meshtastic recommends leaving the hop limit at the default of 3 (maximum 7). Increasing it rarely improves coverage and significantly increases channel congestion; address coverage gaps with better-placed router nodes instead of more hops:

```
# Default is 3; only raise this if you genuinely need it:
meshtastic --set lora.hop_limit 3

# Note: Higher hop limit increases channel utilization and rarely helps coverage.
# Prefer adding well-placed routers over raising the hop count.
```

Ignore MQTT Setting

If your network has an MQTT gateway, setting lora.ignore_mqtt=true makes a node drop LoRa packets that entered the mesh via MQTT, which can reduce duplicate or looped traffic from the gateway:

```
# ignore_mqtt tells the node not to re-transmit over radio any packet that
# arrived from the internet (MQTT), which prevents echo loops:
meshtastic --set lora.ignore_mqtt true
```

Admin Channel Configuration

Set up a separate admin channel for network operations traffic, separate from the public community channel. The ops-channel PSK must be a cryptographically random 256-bit key generated by the app (use `psk random`), not a human-typed string:

```
# Add a second channel for admin use (--ch-add both creates and names it):
meshtastic --ch-add OpsNet
meshtastic --ch-index 1 --ch-set psk random # generates a secure random key; or supply a valid
base64 32-byte key
meshtastic --ch-index 1 --ch-set uplink_enabled false # don't bridge to MQTT
meshtastic --ch-index 1 --ch-set downlink_enabled false
```

Configuration Backup

Always save configuration before deploying. A failed SD card or corrupted firmware update can lose your settings. The export is YAML, not JSON, and is restored with `--configure`:

```
# Export full config to YAML:
meshtastic --export-config > node-config-backup-$(date +%Y%m%d).yaml

# Restore from backup:
meshtastic --configure node-config-backup-20260101.yaml
```