

Running Multiple Rooms

A MeshCore room server is a single node that hosts a **single room**. There is no multi-room mode: one room server node = one room. To offer several separate spaces across a community (for example, a public room and a private emergency operations room), you deploy **multiple room-server nodes** — one node per room. This page covers how a single room server works and how to run several of them as a coordinated set.

How a room server works

A MeshCore room server is a store-and-forward node configured entirely over its serial/BLE CLI — there is **no configuration file**. A single room server has:

- One advertised **node name**, set with `set name <name>` (this is the name clients see; there is one name per node).
- One **admin password**, changed with `password <new>`, used for administrative control of the node.
- One **guest password**, changed with `set guest.password <password>` (default: `hello` — change it immediately). This gates guest access to the room.
- One per-contact **access control list (ACL)**, managed with `setperm <pubkey> <level>` and viewed with `get acl`. Permission levels are Guest (0), Read-only (1), Read-write (2), and Admin (3).
- A single **read-only flag**, `set allow.read.only on|off` (default: **off**). When on, it permits unauthenticated read-only access; by default unauthenticated read is not granted.
- A single persisted **message history** store. The room server retains posts and pushes the unseen backlog to a client when that client next connects (a client receives the previously unseen messages on login — documented as the previous 32). This persistence is the entire point of a room server versus an ordinary channel.

A client connects to a room server (supplying the guest password if one is required) to access that server's single room. There is no "choose which room to join" selection, because a server hosts only one room. A client can, however, hold multiple room servers as contacts and interact with each separately.

Exact password defaults, ACL levels, and the unseen-message count should be verified against your firmware version and the current MeshCore CLI reference at docs.meshcore.io/cli_commands (as of 2026-06-08).

Deploying multiple room servers

Because one node hosts one room, separate spaces require separate nodes. A community that wants a public room and a restricted emergency room runs two room-server nodes, each configured independently over its own CLI. Each node is a separate piece of hardware (for example a Heltec V3 or an nRF52840 board) with its own name, passwords, ACL, and history store.

For example, to set up three room servers across a region you would, on each node in turn, connect over serial/BLE and configure it:

```
# --- Node 1: public regional room ---
set name RegionMesh-Public
password <newAdminPassword>
set guest.password <publicGuestPassword> # or leave blank for open guest access
set allow.read.only on                    # optional: allow unauthenticated read-only
advert

# --- Node 2: restricted emergency room (separate hardware) ---
set name EmergencyNet
password <newAdminPassword>
set guest.password <emergencyGuestPassword>
setperm <coordinatorPubkey> 2             # grant a coordinator Read-write
advert

# --- Node 3: club room (separate hardware) ---
set name ClubNet
password <newAdminPassword>
set guest.password <clubGuestPassword>
advert
```

All of these nodes — like every node in the network — must share the same radio preset (frequency, bandwidth, spreading factor, coding rate) as the companions and repeaters around them; otherwise their packets and adverts cannot be received by other nodes. Use the named region preset in the app or web flasher rather than hand-entering values. Give each node a clear, distinct name so users can tell the rooms apart.

Controlling access to each room

Access to a room is governed by the room server's guest password and per-contact ACL, not by a per-room PSK. Anyone who has a room's guest password can join and read that room's messages, so treat it like a shared secret: distribute it through a separate secure channel (encrypted email, in person, Signal, etc.), not over the mesh itself, and rotate it with `set guest.password <new>` if it may have leaked.

For finer control on a given node, grant individual contacts specific permission levels with `setperm <pubkey> <level>` (Guest/Read-only/Read-write/Admin) and review the current list with `get acl`. For a restricted emergency room, the guest password is typically shared only with vetted local emergency-management personnel, ARES/RACES members, and CERT team leads — not published publicly — and trusted operators can be given Read-write or Admin via the ACL.

Monitoring a room server

A MeshCore room server has no HTTP status endpoint and no management script — it is firmware on a microcontroller (nRF52840/ESP32). You monitor it over the serial/BLE CLI using the built-in stats commands:

```
# System stats: battery, uptime, queue length, debug flags
stats-core

# Radio-layer stats
stats-radio

# Packet-level stats
stats-packets
```

These commands are issued over the serial console (or the companion app's CLI). To monitor several rooms, connect to each room-server node and query it individually.

Revision #7

Created 2026-05-03 03:52:11 UTC by Mesh America Admin

Updated 2026-06-10 05:25:17 UTC by Mesh America Admin