



☰ macOS companion app > Remote Control

macOS companion app

Remote Control

This flow lets the macOS app act as a full remote control for a OpenClaw gateway running on another host (desktop/server). It's the app's **Remote over SSH** (remote run) feature. All features—health checks, Voice Wake forwarding, and Web Chat—reuse the same remote SSH configuration from *Settings* → *General*.

Modes

Local (this Mac): Everything runs on the laptop. No SSH involved.

Remote over SSH (default): OpenClaw commands are executed on the remote host. The mac app opens an SSH connection with `-o BatchMode` plus your chosen identity/key and a local port-forward.

Remote direct (ws/wss): No SSH tunnel. The mac app connects to the gateway URL directly (for example, via Tailscale Serve or a public HTTPS reverse proxy).

Remote transports

Remote mode supports two transports:

SSH tunnel (default): Uses `ssh -N -L ...` to forward the gateway port to localhost. The gateway will see the node's IP as `127.0.0.1` because the tunnel is loopback.



Direct (ws/wss): Connects straight to the gateway URL. The gateway sees the real client IP.

>

Prereqs on the remote host

1. Install Node + pnpm and build/install the OpenClaw CLI (`pnpm install && pnpm build && pnpm link --global`).
2. Ensure `openclaw` is on PATH for non-interactive shells (symlink into `/usr/local/bin` or `/opt/homebrew/bin` if needed).
3. Open SSH with key auth. We recommend **Tailscale** IPs for stable reachability off-LAN.

macOS app setup

1. Open *Settings* → *General*.
2. Under **OpenClaw runs**, pick **Remote over SSH** and set:

Transport: `SSH tunnel` or `Direct (ws/wss)`.

SSH target: `user@host` (optional `:port`).

If the gateway is on the same LAN and advertises Bonjour, pick it from the discovered list to auto-fill this field.

Gateway URL (Direct only): `wss://gateway.example.ts.net` (or `ws://...` for local/LAN).

Identity file (advanced): path to your key.

Project root (advanced): remote checkout path used for commands.

CLI path (advanced): optional path to a runnable `openclaw` entrypoint/binary (auto-filled when advertised).

3. Hit **Test remote**. Success indicates the remote `openclaw status --json` runs correctly. Failures usually mean PATH/CLI issues; exit 127 means the CLI isn't found remotely.



4. Health checks and Web Chat will now run through this SSH tunnel automatically.

>

Web Chat

SSH tunnel: Web Chat connects to the gateway over the forwarded WebSocket control port (default 18789).

Direct (ws/wss): Web Chat connects straight to the configured gateway URL.

There is no separate WebChat HTTP server anymore.

Permissions

The remote host needs the same TCC approvals as local (Automation, Accessibility, Screen Recording, Microphone, Speech Recognition, Notifications). Run onboarding on that machine to grant them once.

Nodes advertise their permission state via `node.list` / `node.describe` so agents know what's available.

Security notes

Prefer loopback binds on the remote host and connect via SSH or Tailscale.

If you bind the Gateway to a non-loopback interface, require token/password auth.

See [Security](#) and [Tailscale](#).

WhatsApp login flow (remote)

Run `openclaw channels login --verbose` **on the remote host**. Scan the QR with WhatsApp on your phone.



Re-run login on that host if auth expires. Health check will surface link problems.

>

Troubleshooting

exit 127 / not found: `openclaw` isn't on PATH for non-login shells. Add it to `/etc/paths`, your shell rc, or symlink into `/usr/local/bin` / `/opt/homebrew/bin`.

Health probe failed: check SSH reachability, PATH, and that Baileys is logged in (`openclaw status --json`).

Web Chat stuck: confirm the gateway is running on the remote host and the forwarded port matches the gateway WS port; the UI requires a healthy WS connection.

Node IP shows 127.0.0.1: expected with the SSH tunnel. Switch **Transport** to **Direct (ws/wss)** if you want the gateway to see the real client IP.

Voice Wake: trigger phrases are forwarded automatically in remote mode; no separate forwarder is needed.

Notification sounds

Pick sounds per notification from scripts with `openclaw` and `node.invoke`, e.g.:

```
openclaw nodes notify --node <id> --title "Ping" --body "Remote gateway"
```

There is no global “default sound” toggle in the app anymore; callers choose a sound (or none) per request.



Powered by mintlify

>