



☰ Configuration and operations > **Gateway Lock**

Configuration and operations

Gateway Lock

Last updated: 2025-12-11

Why

Ensure only one gateway instance runs per base port on the same host; additional gateways must use isolated profiles and unique ports.

Survive crashes/SIGKILL without leaving stale lock files.

Fail fast with a clear error when the control port is already occupied.

Mechanism

The gateway binds the WebSocket listener (default `ws://127.0.0.1:18789`) immediately on startup using an exclusive TCP listener.

If the bind fails with `EADDRINUSE`, startup throws `GatewayLockError("another gateway instance is already listening on ws://127.0.0.1:<port>")`.

The OS releases the listener automatically on any process exit, including crashes and SIGKILL—no separate lock file or cleanup step is needed.

On shutdown the gateway closes the WebSocket server and underlying HTTP server to free the port promptly.

>

Error surface

If another process holds the port, startup throws
`GatewayLockError("another gateway instance is already listening on ws://127.0.0.1:<port>")`.

Other bind failures surface as `GatewayLockError("failed to bind gateway socket on ws://127.0.0.1:<port>: ...")`.

Operational notes

If the port is occupied by *another* process, the error is the same; free the port or choose another with `openclaw gateway --port <port>`.

The macOS app still maintains its own lightweight PID guard before spawning the gateway; the runtime lock is enforced by the WebSocket bind.

< Logging

Background Exec and Process Tool >

Powered by [mintlify](#)