OPENCLAW DESKTOP COMPANION

OCC Desktop

User Manual

Mission Control for Your Bridge Crew

Version: v0.3.50 **Date:** May 2026 **Platform:** Linux Desktop

Monitor your OpenClaw gateway, coordinate your Bridge Crew, and manage sessions from a dedicated desktop command center.

Support: `support@occdesktop.com`

This is the editable HTML source for the OCC Desktop production manual PDF. It reconstructs the original manual content and incorporates the validated v0.3.50 install, pairing, uninstall, and reinstall flow.

Table of Contents

0. Before You Start

Before installing OCC Desktop, make sure you have:

- OpenClaw Gateway running and reachable from this machine
- Gateway host address (IP or hostname)
- Gateway token available (see **How to Obtain Your Gateway Token**)
- Network path working (check with `ping <gateway-host>`)

> **Pro tip:** The gateway default port is `18789` and the metrics default is `18790`. If you're using defaults, your connection should be straightforward.

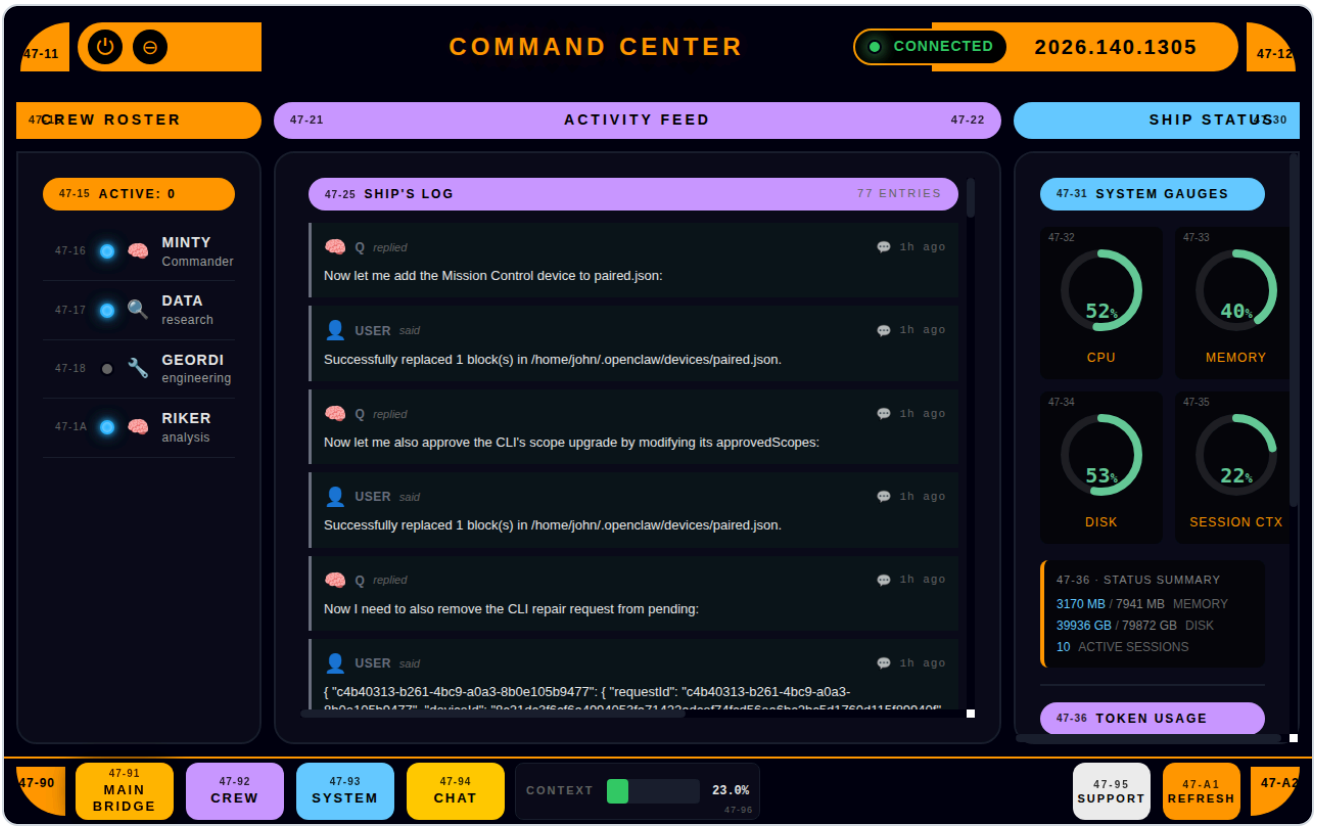
1. Introduction

What is OCC Desktop?

OCC Desktop (OpenClaw Command Center) is an Electron-based desktop application for monitoring and managing your OpenClaw environment.

Use OCC Desktop to:

- Monitor gateway health, active sessions, and host metrics
- See what your Bridge Crew (subagents) are doing in real time
- Track token usage and diagnostics per session
- Chat directly with your primary agent from a dedicated Command Chat view
- Manage crew definitions (names, roles, models, emoji) and pairing state



OCC Desktop provides a dedicated desktop command center for your OpenClaw environment.

OpenClaw is the system that runs your agent(s), manages sessions, and exposes operational APIs through the OpenClaw Gateway.

At a high level:

- The Gateway is the control plane (sessions, routing, auth)
- One or more agents handle user conversations and tasks
- Subagents (your Bridge Crew) can be spawned to do specialized work

What is a Bridge Crew?

A Bridge Crew is a named set of specialist subagents you can spawn from your main agent. OCC Desktop treats them like an operational team:

- You can see who is active, what model they're running, and how far along they are
- You can keep their roles consistent across sessions

> **Pro tip:** Consistency matters. In OpenClaw and OCC Desktop, crew member names and labels must match exactly across your setup, including capitalization. If a crew member doesn't show up, it's usually a label mismatch.

System Requirements

Supported OS: Linux (tested on Linux Mint)
App: OCC Desktop v0.3.50 (Electron)
Backend: OpenClaw Gateway reachable over network (LAN, VPN, or tailnet)

You will need:

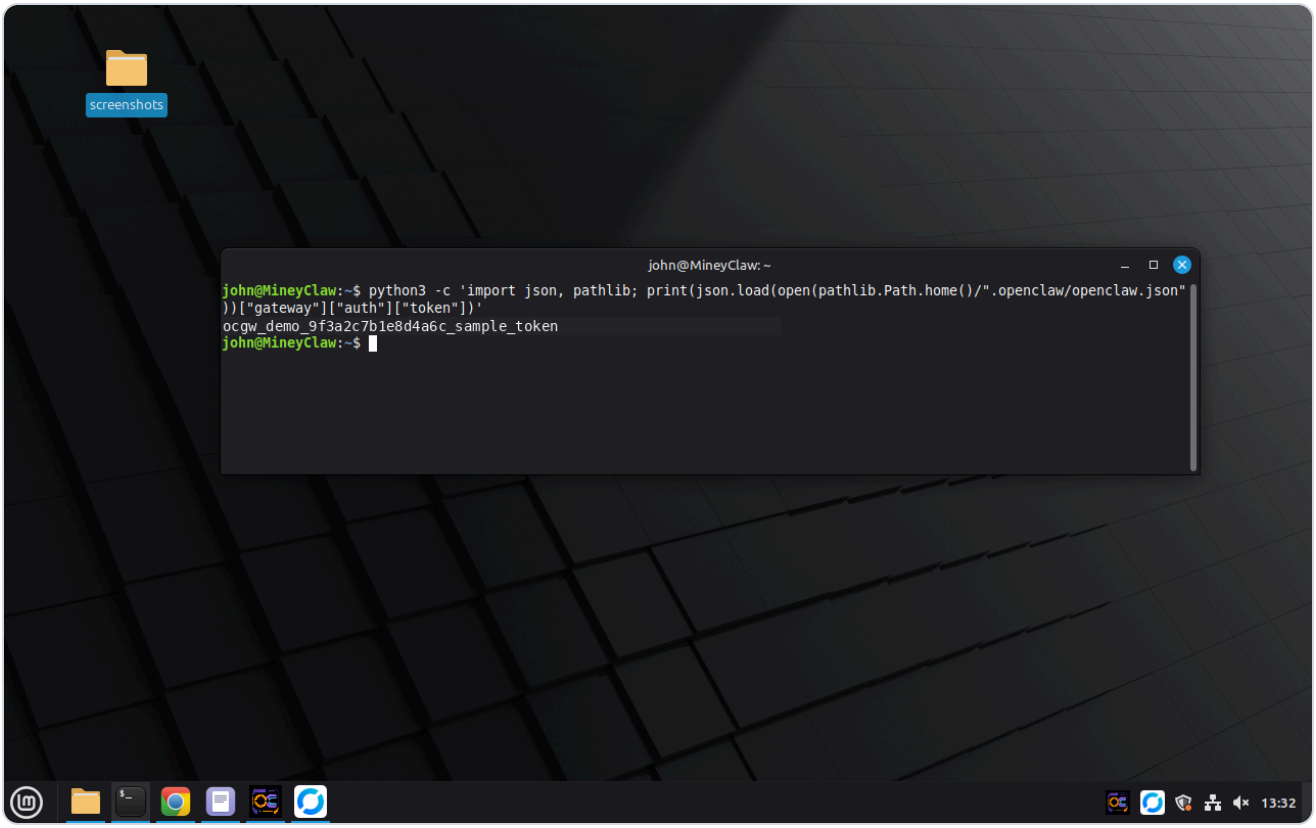
- The OpenClaw Gateway host address (IP or hostname)
- The Gateway WebSocket port (default: 18789)
- Your Gateway token
- The Gateway protocol (`ws` for local, `wss` for remote with TLS)
- A metrics endpoint URL for host gauges (for example `http://gateway-host:18790`)

How to Obtain Your Gateway Token

Your Gateway token is required for OCC Desktop to authenticate with the OpenClaw Gateway.

If you run OpenClaw yourself (self-hosted)

```
python3 -c 'import json, pathlib; print(json.load(open(pathlib.Path.home()/".op
```



Run this command on the machine where OpenClaw is installed to print the current gateway token.

This prints your current operator token. If your deployment uses a custom token location, check your OpenClaw configuration under `gateway.auth.token`.

If someone else manages your gateway

Ask your OpenClaw administrator for:

- the gateway host address
- the token string
- confirmation that your machine is allowed to connect

> **Important:** Gateway tokens grant operational access. Do not share them in screenshots, chat logs, or public documentation. If a token is exposed, rotate it immediately.

2. Installation

OCC Desktop is distributed as a self-contained `.run` installer built with Qt Installer Framework (Qt IFW).

Downloading the Installer

1. Obtain the OCC Desktop `.run` installer from your release location.
2. Copy it to your Linux machine. Your Downloads folder is fine.

Running the Installer (Qt IFW)

1. Make the installer executable:

```
chmod +x OCC-Desktop-Installer-v0.3.50.run
```

2. Run it:

```
./OCC-Desktop-Installer-v0.3.50.run
```

3. Follow the Qt IFW wizard:

- accept license if presented
- confirm installation path (default: `/opt/occd/`)
- complete installation

Installation Locations

Default installation path: `/opt/occd/`

Contents include:

- application files under `/opt/occd/`
- desktop launcher entry in your application menu

Uninstalling OCC Desktop

To fully remove OCC Desktop, use the installed maintenance tool located at:

```
/opt/occd/maintenancetool
```

Launch it and choose **Uninstall**.

This is the recommended removal method because it performs a full OCC Desktop uninstall, including:

- application files under `/opt/occd`
- local OCC Desktop configuration
- local OCC Desktop cache
- desktop and menu integration entries

After uninstall, reinstalling OCC Desktop should behave like a fresh install and show onboarding again.

> **Pro tip:** Use the maintenance tool rather than deleting directories manually. In v0.3.50, uninstall cleanup was validated and now removes the app, config, cache, and desktop entries properly.

3. First Launch & Onboarding

On first launch, OCC Desktop guides you through a 4-step Setup Wizard. This gets your gateway connection and crew definitions aligned with how OpenClaw expects them.

Step 1: Welcome

You'll see a quick introduction to OCC Desktop and what it can monitor.

Step 2: Connection (TEST GATEWAY)

Configure how OCC Desktop reaches your OpenClaw Gateway.

Field	Description	Example
Gateway Host	IP address or hostname of the machine running OpenClaw	`192.168.1.12` or `gateway.local`
Gateway Port	WebSocket port (nearly always the default)	`18789`
Protocol	`ws` for local networks, `wss` for TLS-secured remote connections	`ws`
Metrics Base URL	Full HTTP URL to the metrics server	`http://192.168.1.12:18790`
Gateway Token	Your operator token — required	See token section above

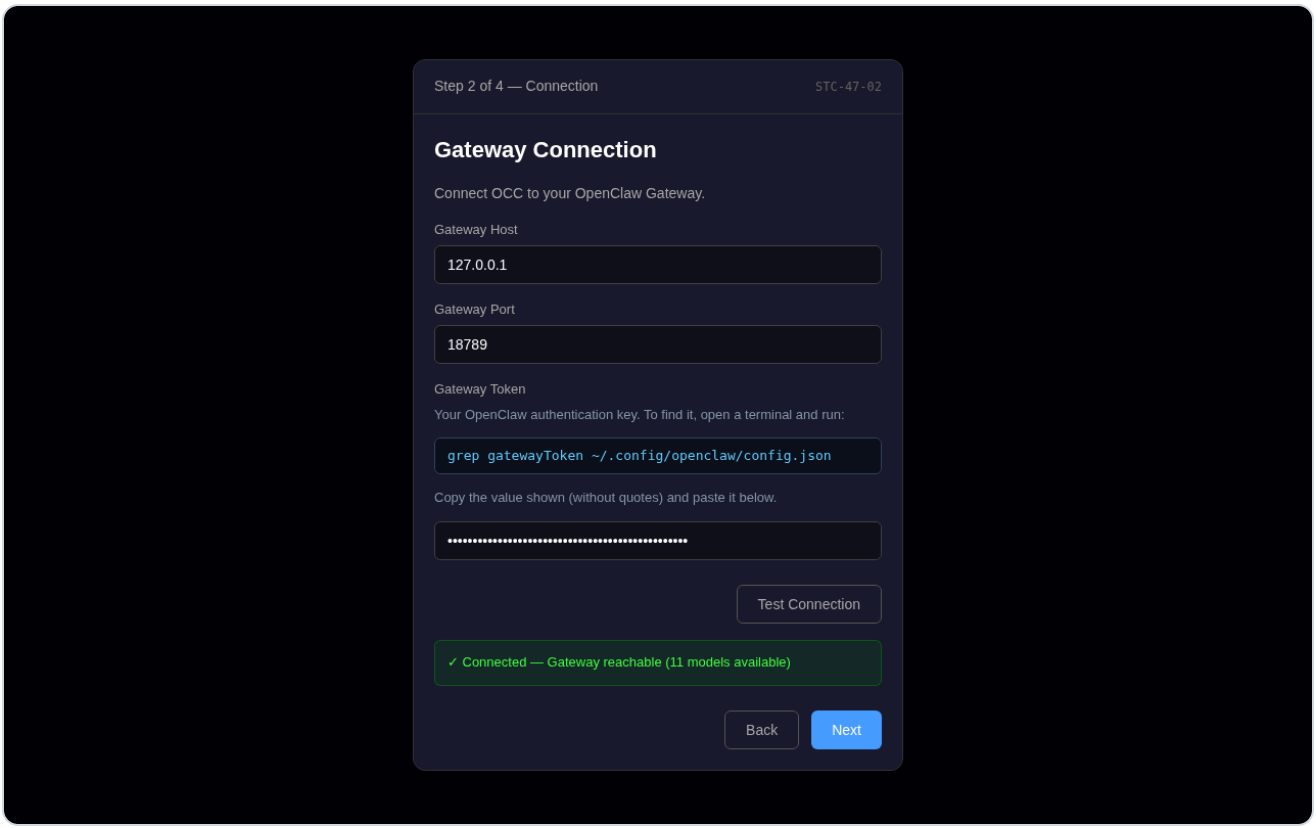
> **Important:** Gateway Host, Gateway Port, Protocol, and Gateway Token are required. Metrics Base URL is strongly recommended — without it, host gauges will not display.

After filling in all fields, click **TEST GATEWAY** to verify:

- OCC can reach the gateway over the network

- the token is valid and accepted
- session and metrics data can be retrieved

If the test succeeds, you'll see a green confirmation banner.



The connection step of the Setup Wizard, where gateway details are entered and tested.

Pro tip: If TEST GATEWAY fails, first verify the host is reachable with `ping gateway-host`, then confirm the token is current with `python3 -c 'import json, pathlib; print(json.load(open(pathlib.Path.home()/".openclaw/openclaw.json"))["gateway"]["auth"]["token"])'`.

Step 3: Crew Setup (roles, models, emoji)

Define your Bridge Crew so OCC can display them consistently.

Each crew member needs:

- **Name/Label** — must match exactly what OpenClaw uses when spawning subagents (case sensitive)

- **Role** — what they're responsible for
- **Model** — the model identifier they should use (must be allowed in your OpenClaw config)

Suggested crew pattern (current example):

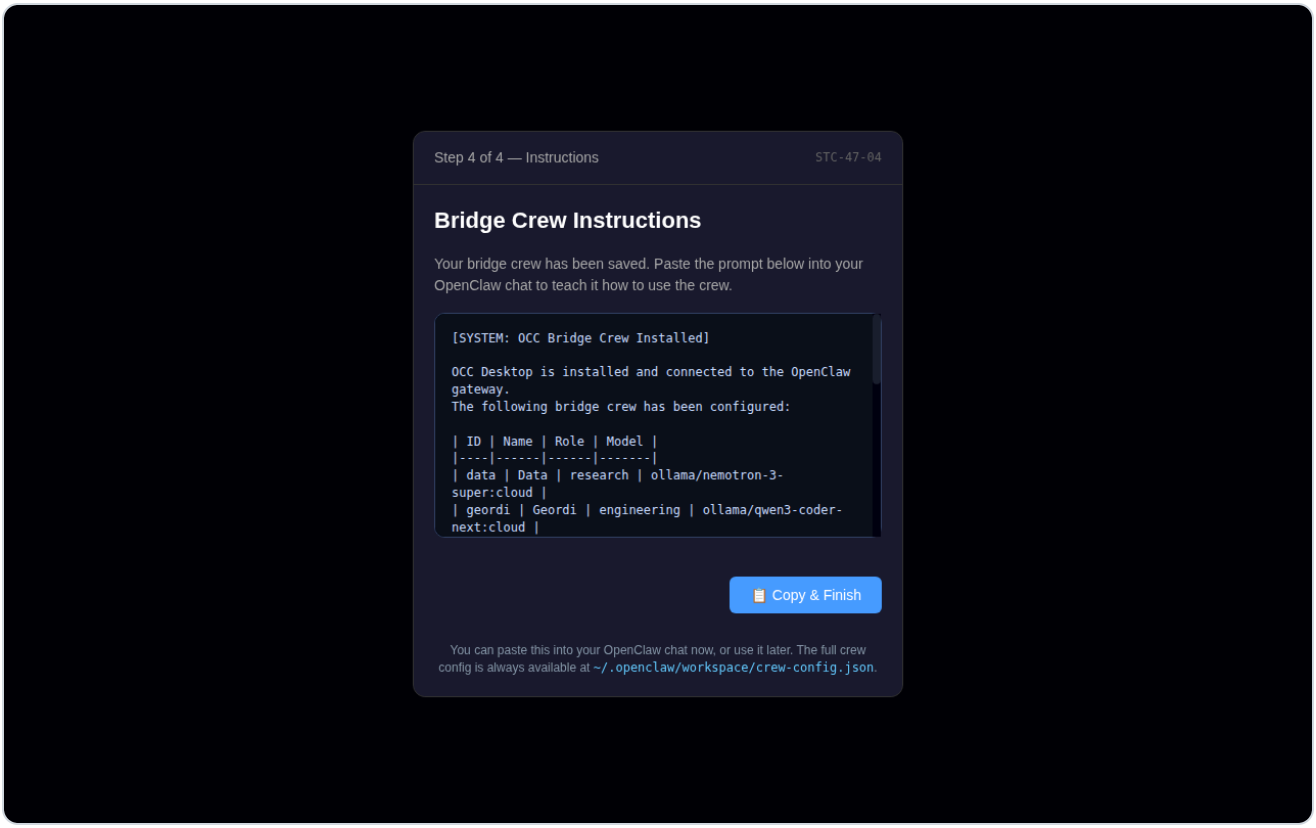
Crew Member	Role	Model (example)
Data	Research & Analysis	`openai-codex/gpt-5.3-codex`
Geordi	Backend Engineering	`openai-codex/gpt-5.3-codex`
Spark	Frontend / UI	`openai-codex/gpt-5.3-codex-spark`
Riker	QA & Review	`openai-codex/gpt-5.4-mini`
Troi	Copy / User-Facing Text	`openai-codex/gpt-5.2`
Barclay	Art & Design	`google/gemini-3.1-flash-image-preview`

Step 4: Bridge Crew Instructions (copiable prompt)

The wizard provides a copyable instruction block to paste into your OpenClaw agent's system or developer instructions. Click **Copy & Finish** to copy the instructions to your clipboard, then paste them into your agent configuration.

This ensures OCC Desktop and OpenClaw agree on:

- which crew members exist
- what each crew member is called
- which models each is allowed to use



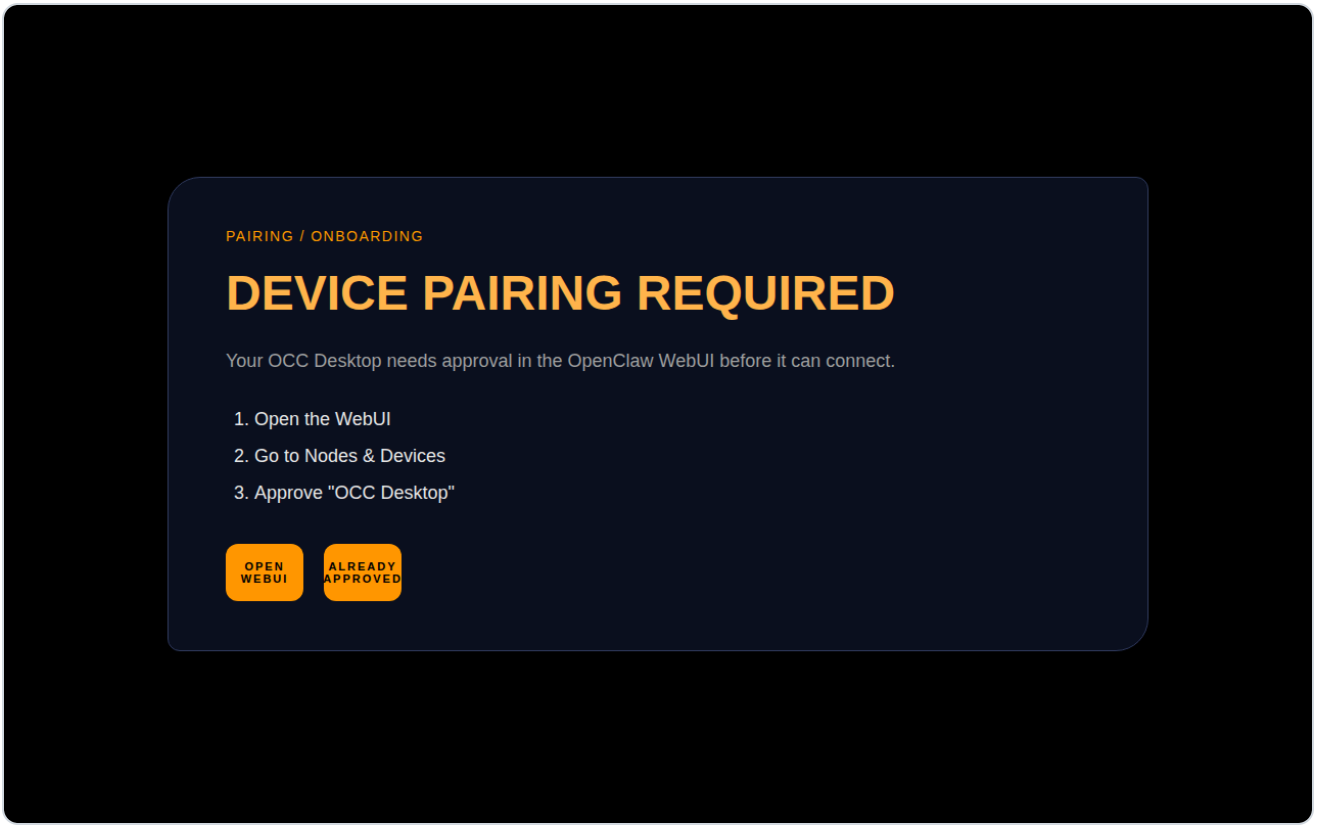
The final onboarding step generates a copyable bridge crew instruction block.

Pairing Approval After Onboarding

In v0.3.50, if the gateway requires device approval, OCC Desktop no longer drops you into a confusing failed state. Instead, it shows a dedicated pairing screen immediately after onboarding.

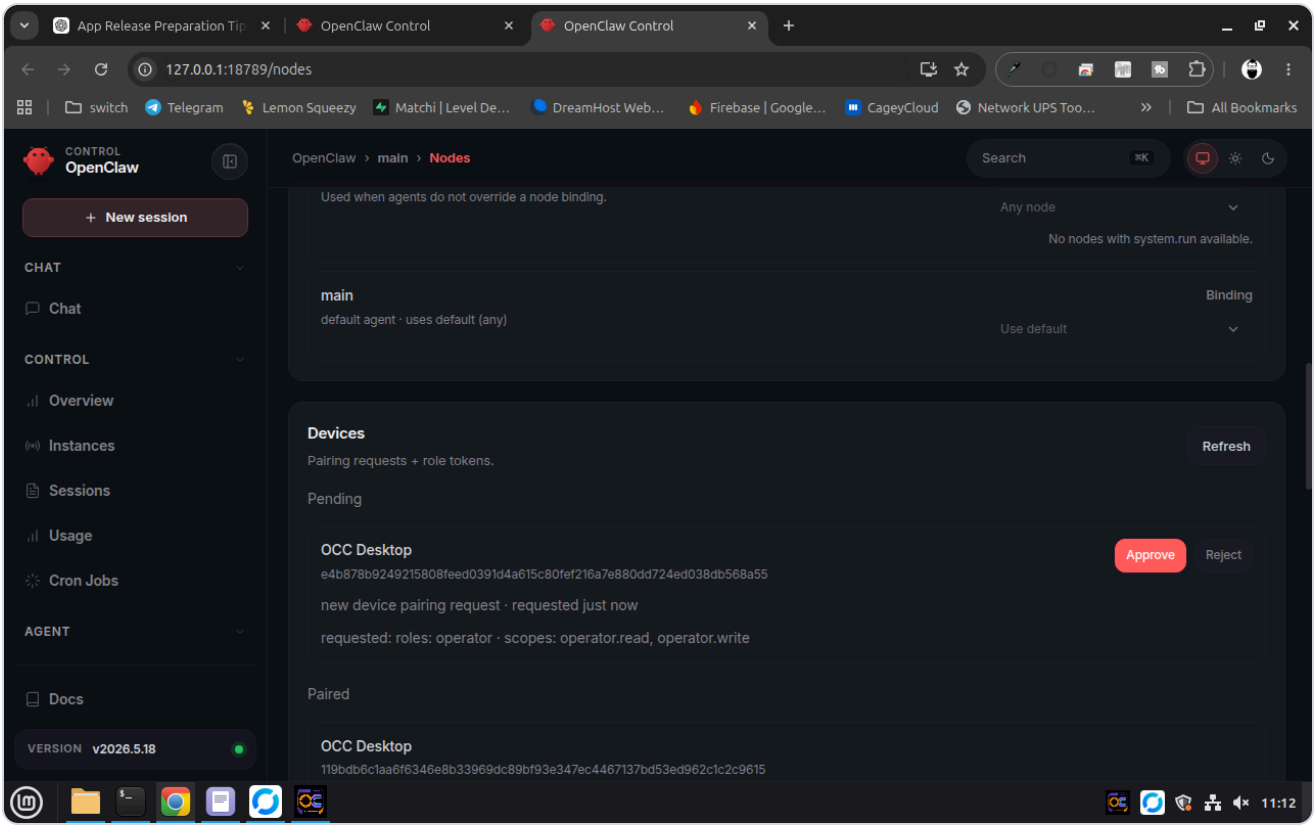
This pairing screen gives you two actions:

- **Open WebUI** — opens the OpenClaw WebUI directly to the `/nodes` page
- **Already Approved** — retries the gateway connection after the device has been approved



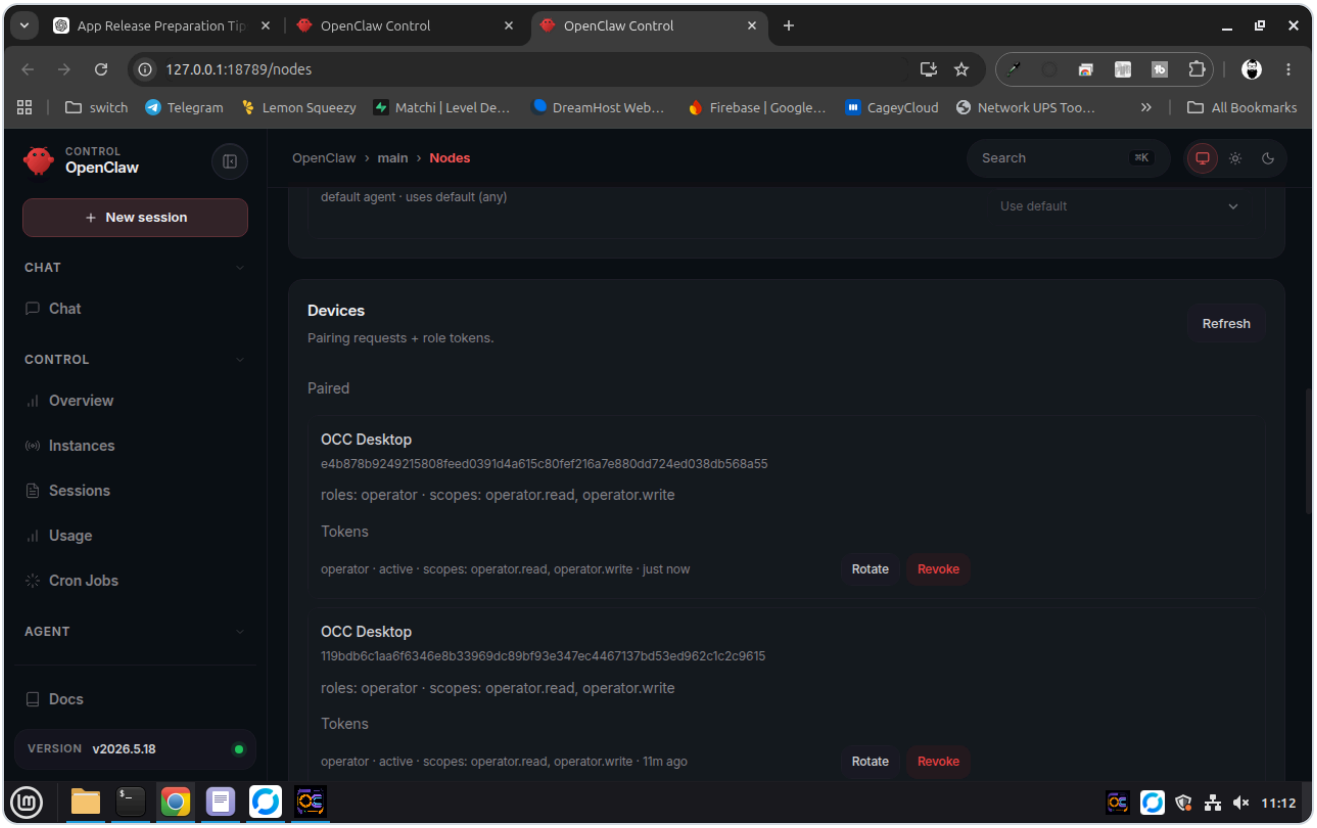
The full-screen pairing interstitial shown when the device still needs approval.

After clicking **Open WebUI**, go to **Nodes & Devices** and approve the pending **OCC Desktop** device request.



The pending OCC Desktop pairing request in OpenClaw WebUI.

Once approved, return to OCC Desktop and click **Already Approved**. The app should reconnect and continue to the main screen.

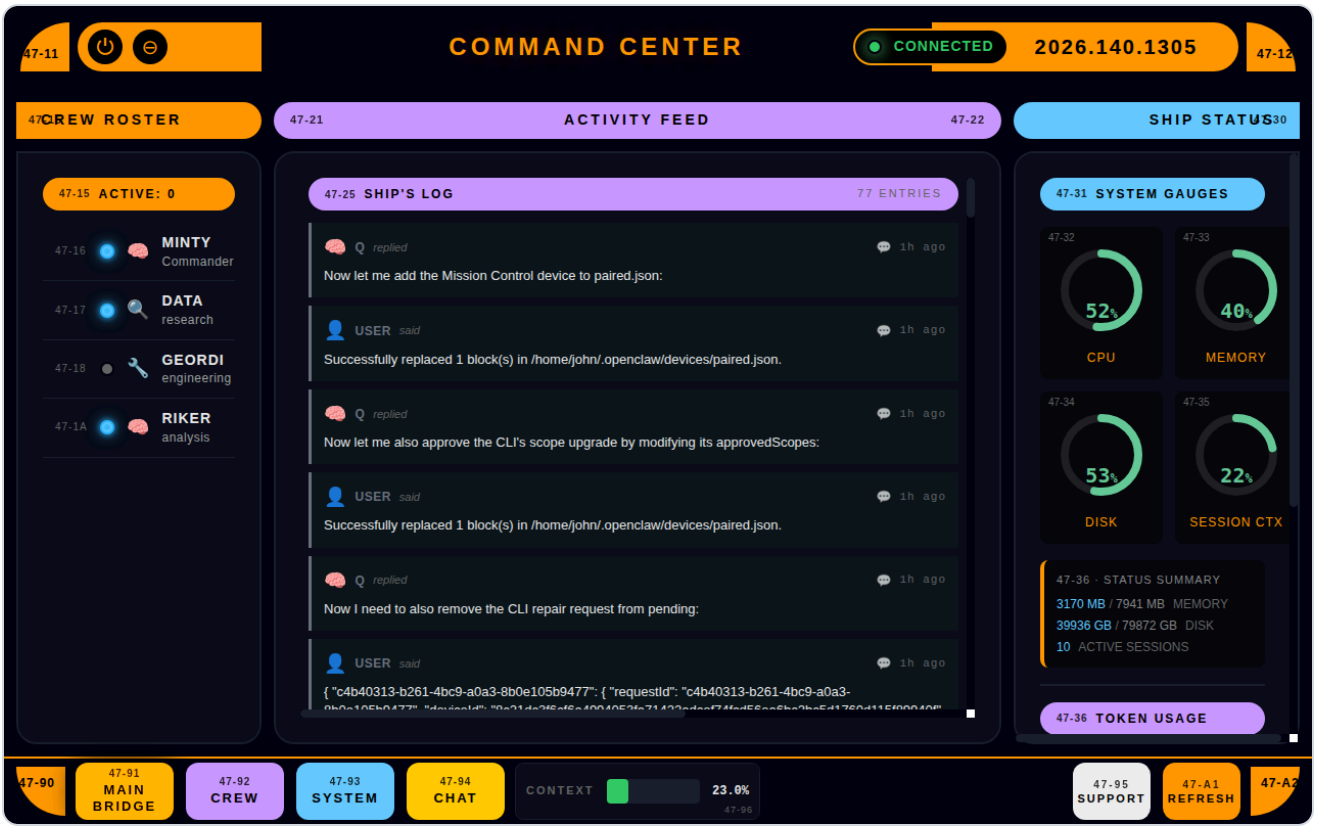


The OCC Desktop device after approval has been granted.

Pro tip: If the approval has been granted but the app still hasn't resumed, click **Already Approved** again or relaunch OCC Desktop and reconnect.

4. Main Bridge View

The Main Bridge is the operational dashboard: a fast overview of crew activity, event history, and system status.



The Main Bridge view combines crew activity, system status, and the current bottom navigation layout.

Header and Layout

The Main Bridge view is organized into three columns:

- **Left: Crew Roster** — active or idle state, online or offline indicator, quick identity
- **Center: Activity Feed (Ship's Log)** — session creation, pairing events, connection status, warnings, errors
- **Right: Ship Status / Host Metrics** — CPU, memory, and disk usage

> **Pro tip:** Treat the gauges as early warning, not a postmortem. If CPU or memory trends high during heavy agent work, consider limiting concurrent sessions.

The header also surfaces quick status signals:

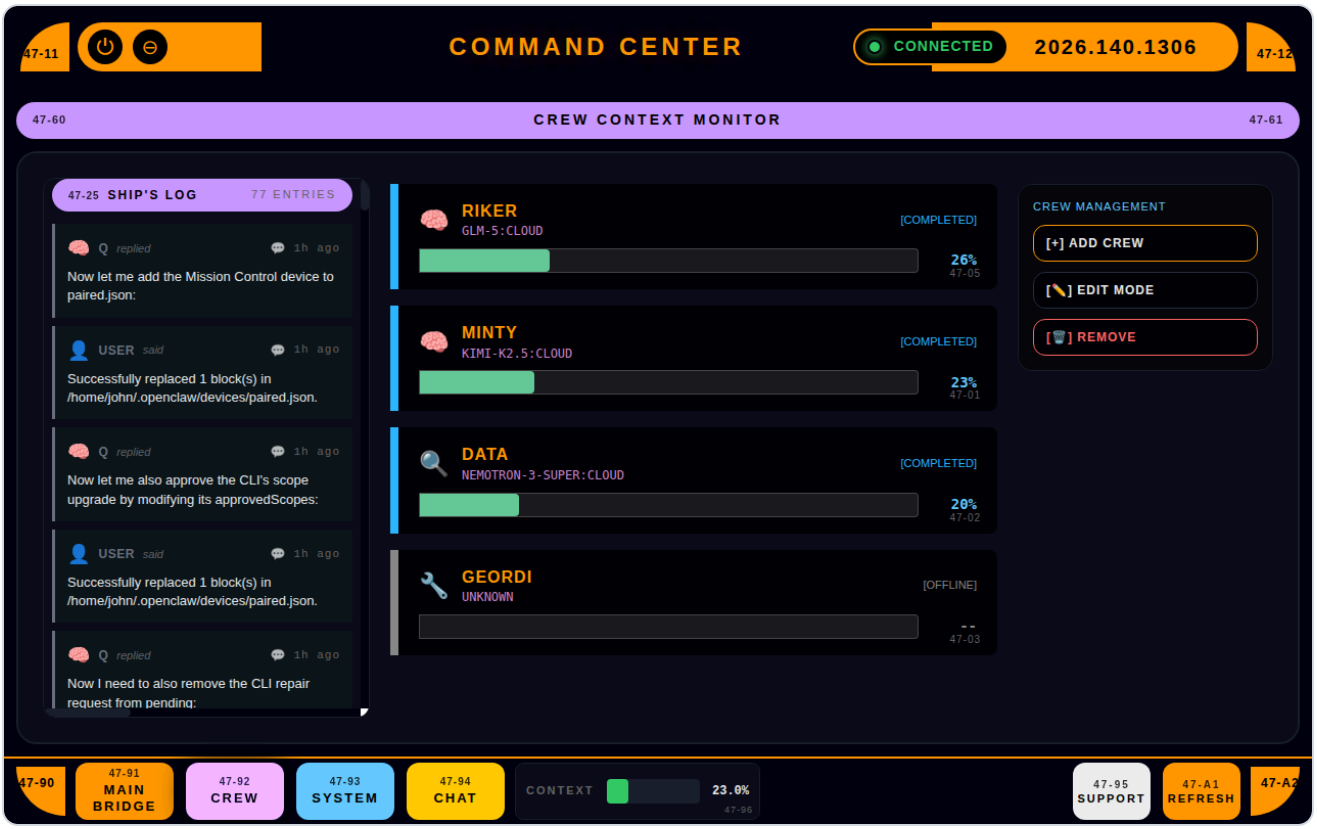
- connected or disconnected state
- gateway identity and version
- pairing pending badge when applicable

Bottom Navigation Bar

The bottom navigation switches between the major OCC Desktop sections. In the current interface, the exact buttons and labels may evolve as the product grows, but the core operational views include the Main Bridge, Crew, System, Chat, Support, and Settings-related actions.

5. Crew Management

Crew Management is where you inspect, add, and maintain your Bridge Crew definitions and view live subagent progress.



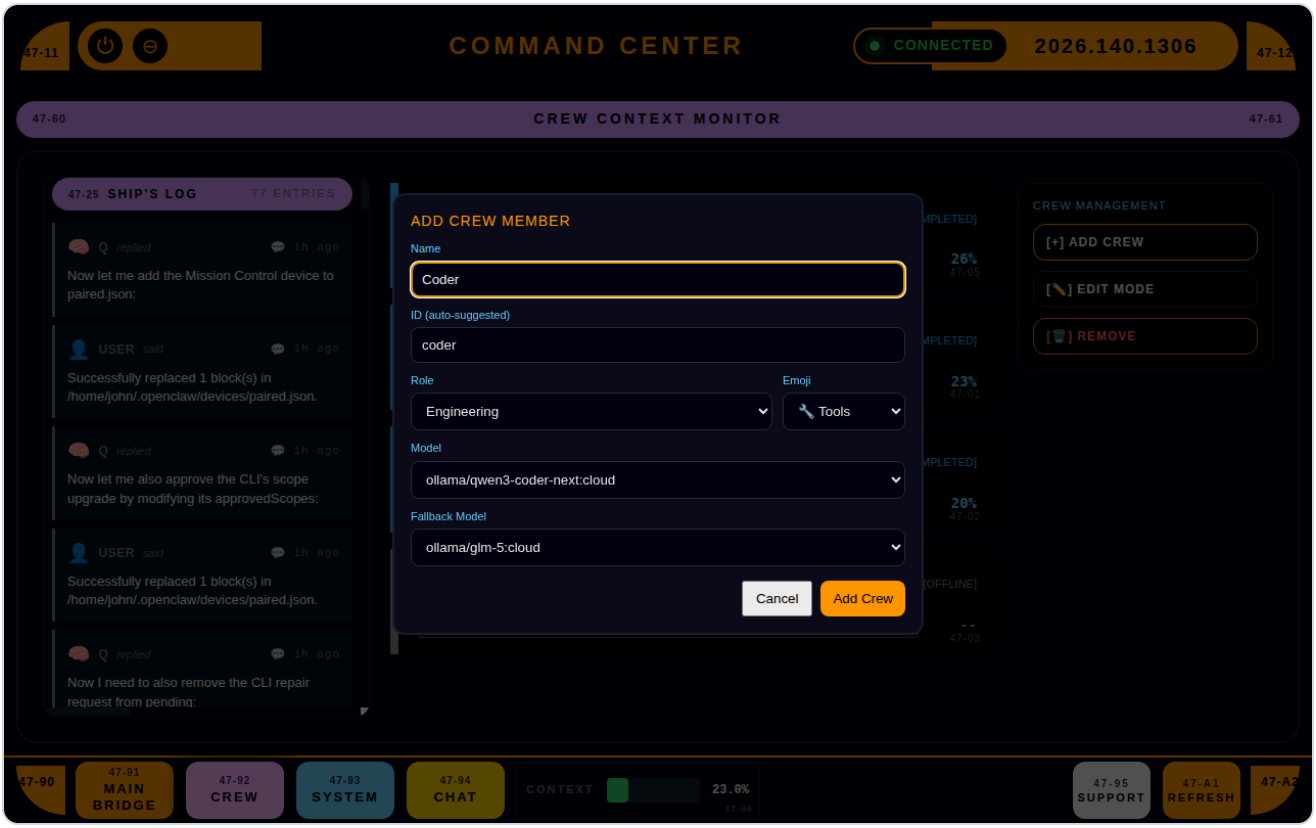
Crew Management shows configured crew members, their roles, and current operational state.

Viewing Your Bridge Crew

The crew list shows all configured crew members, including those currently offline. Each entry shows crew name, role summary, current model when known, and online or offline status.

Crew Management (progress cards)

Per-crew progress cards show who is currently working, approximate progress percentage, whether a crew member is offline, and which model is in use when available.



The Add Crew Member view is used to define new crew entries and keep OCC aligned with your OpenClaw setup.

Adding New Crew Members

Use **Add Crew** to define an additional subagent. Fill in:

- Name/Label — must match what the agent uses when spawning
- Role — a clear, single-purpose description
- Model — ensure it's allowed by your OpenClaw model configuration

Editing Crew Members

Update an existing crew member's role, model, or emoji. Common edits include tightening role wording, swapping to a cheaper or faster model for routine tasks, or changing emoji for quick visual scanning.

Removing Crew Members

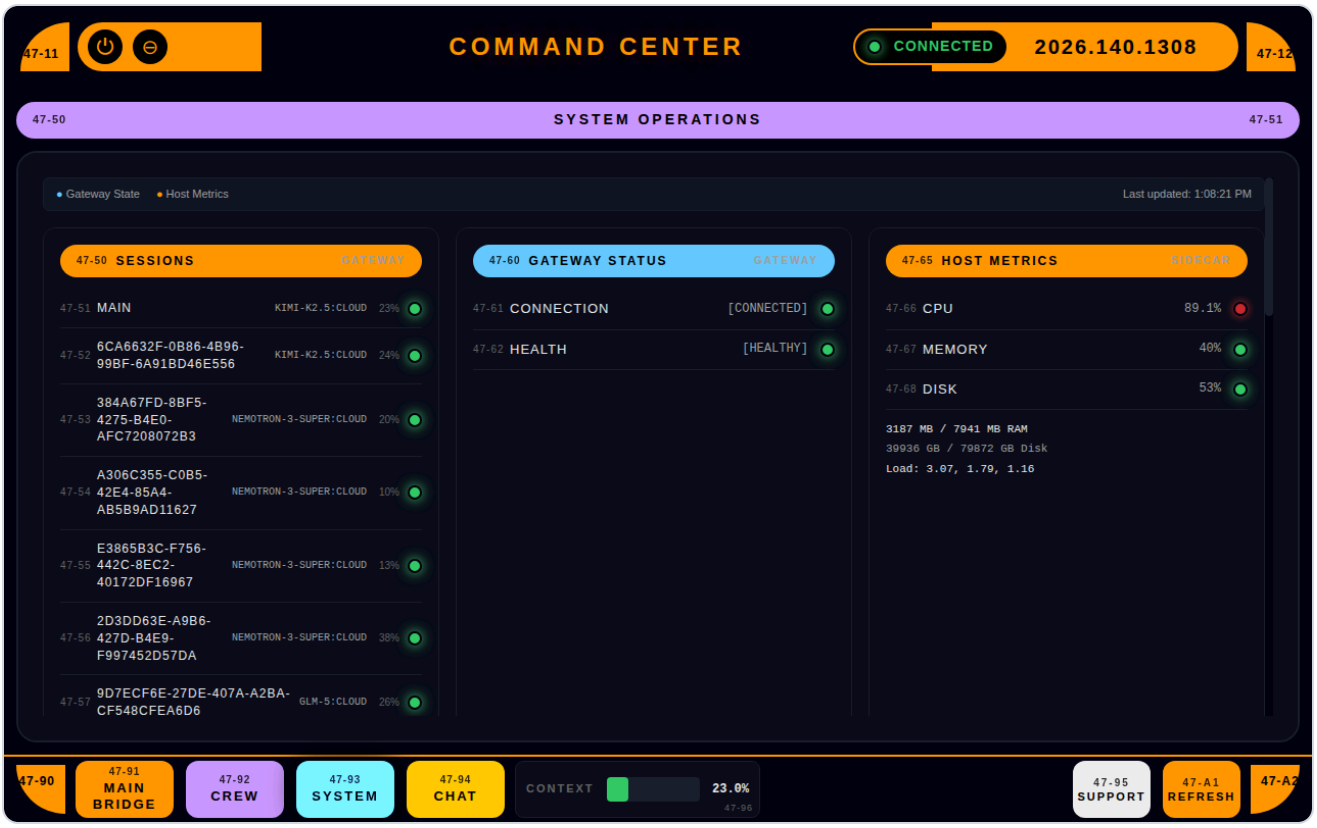
Remove crew members you no longer use to keep monitoring clean.

Pro tip: If you remove a crew member in OCC but your agent still tries to spawn them, you'll get errors. Keep OCC and agent instructions aligned after any changes.

Some environments require nodes or devices to be paired before they can participate. See **Pairing Nodes** for full details.

6. System Operations

System Operations focuses on the health of the OpenClaw Gateway and the host machine.



System Operations focuses on gateway state, metrics, and operational visibility.

Gateway State View

Shows active sessions with IDs, health status, and session counts. Use it to confirm sessions exist when crew are active and to spot orphaned sessions or unexpected spikes.

Host Metrics

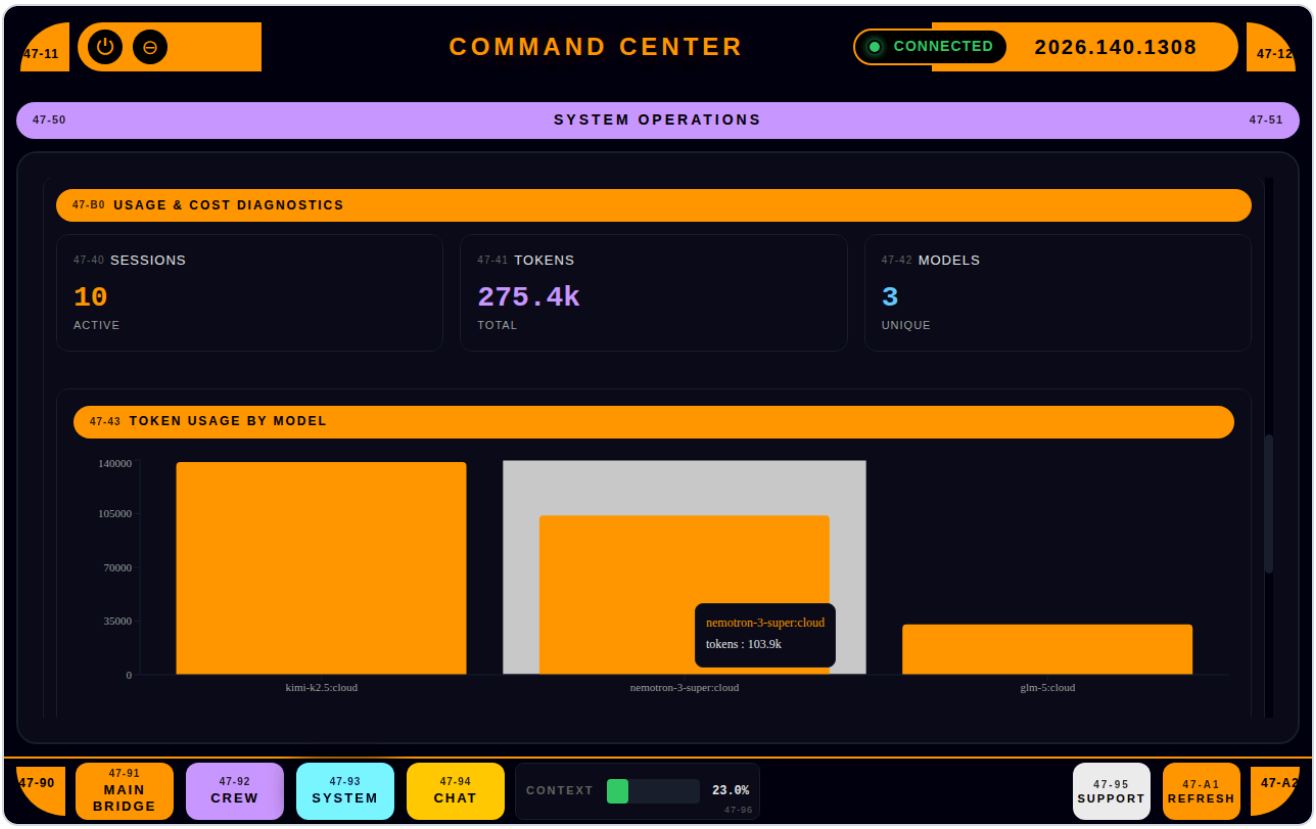
When your Metrics Base URL is configured, OCC displays CPU, memory, and disk gauges. These help you answer whether the gateway host is under pressure and whether the current models are saturating CPU or RAM.

Channels and Version Info

Displays enabled channels and integrations, plus gateway version information. Useful when troubleshooting mismatched client and server versions.

7. Usage & Cost Tracking

OCC Desktop includes usage diagnostics to help you understand token consumption and session distribution.



The token usage view helps identify token-heavy sessions and model cost patterns.

The diagnostics panel shows:

- active sessions count
- total tokens consumed for the period
- unique models in use
- token distribution chart

Per Session Breakdown

Drill into individual sessions to see token usage per session. Use this to identify which sessions are consuming the most tokens, which tasks or models drive costs, and whether a runaway session needs intervention.

Controlling Usage

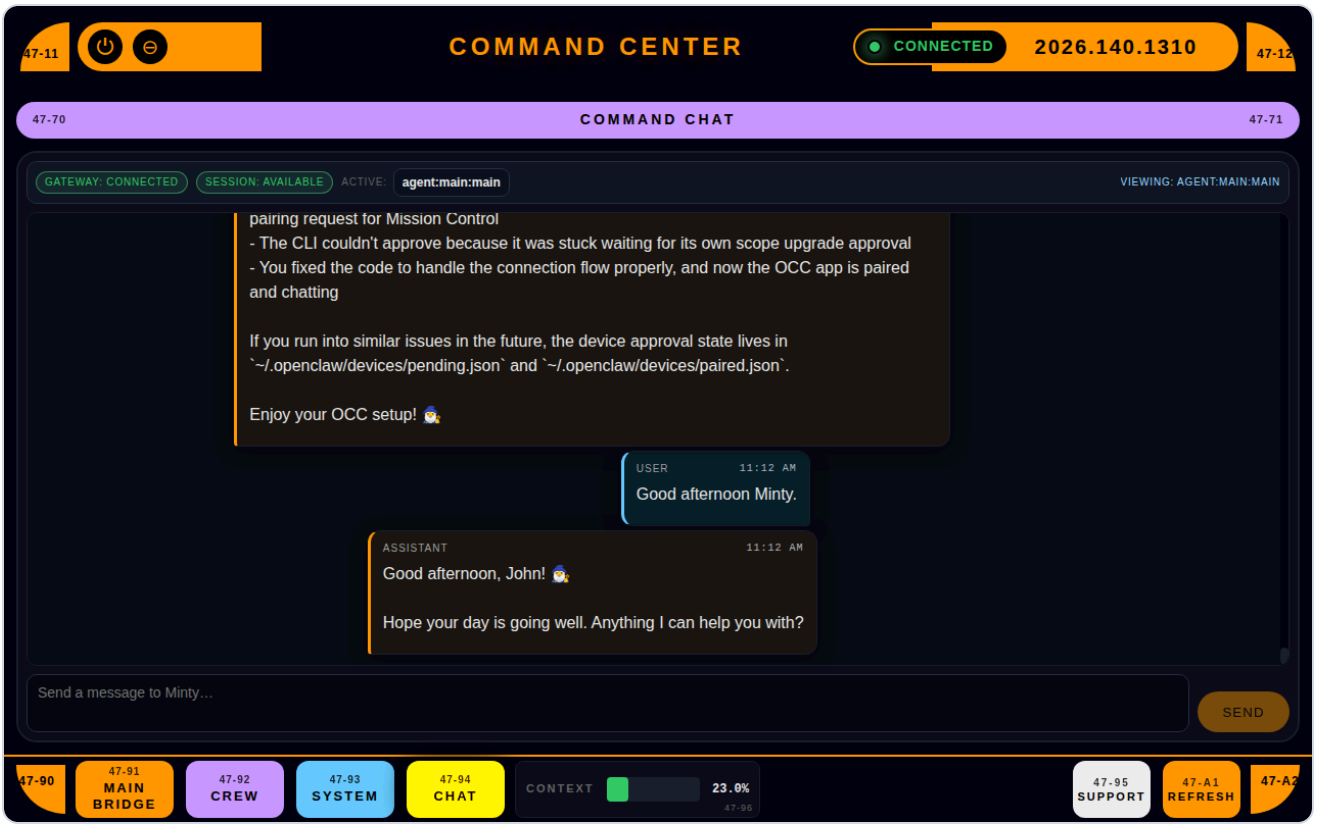
Practical strategies:

- use cheaper or faster models for routine tasks
- limit parallel subagents when concurrency isn't needed
- keep prompts short and specific to reduce iterative churn

Pro tip: If a single session dominates token usage, open Command Chat and ask the agent for a status check. A quick redirect can save significant burn.

8. Command Chat

Command Chat lets you talk directly to your OpenClaw agent from inside OCC Desktop.



Command Chat provides a dedicated in-app conversation view for directing your agent.

Session Selection

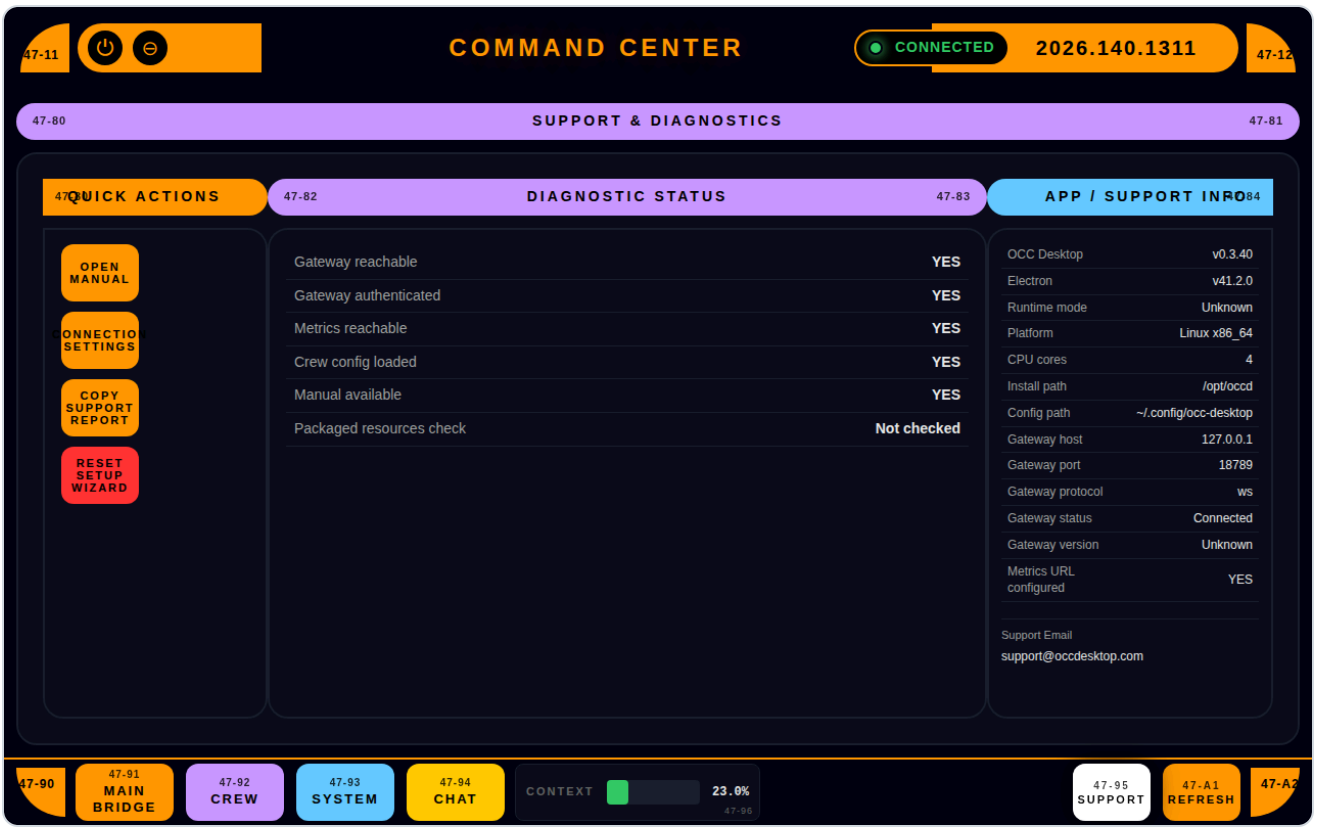
If multiple sessions are active, select the session you want to address. Best practice: use one primary session for ongoing direction and keep specialized sessions separate.

Using Command Chat

Command Chat is the in-app conversation view for directing your primary OpenClaw agent. Use it when you want to issue instructions, ask for status, or coordinate active work without leaving OCC Desktop.

9. Support & Diagnostics

The Support view brings together quick actions, connection diagnostics, connection settings, version details, and support-oriented reference information in one place. It is the best place to start when something looks wrong, when you need app details, or when you want to collect information before troubleshooting.



The Support & Diagnostics page provides app info, quick actions, and troubleshooting-oriented status details.

Left Panel: Quick Actions

The left panel contains the operational shortcuts for support tasks:

- **Open Manual** — opens the OCC Desktop user manual.
- **Connection Settings** — opens the gateway connection settings so you can update host, port, protocol, metrics URL, or token values.
- **Copy Support Report** — copies diagnostics information so you can paste it into an email to `support@occdesktop.com`.

- **Reset Setup Wizard** — resets onboarding so OCC Desktop behaves like a first-run app again without requiring a reinstall.

Center Panel: Diagnostic Status

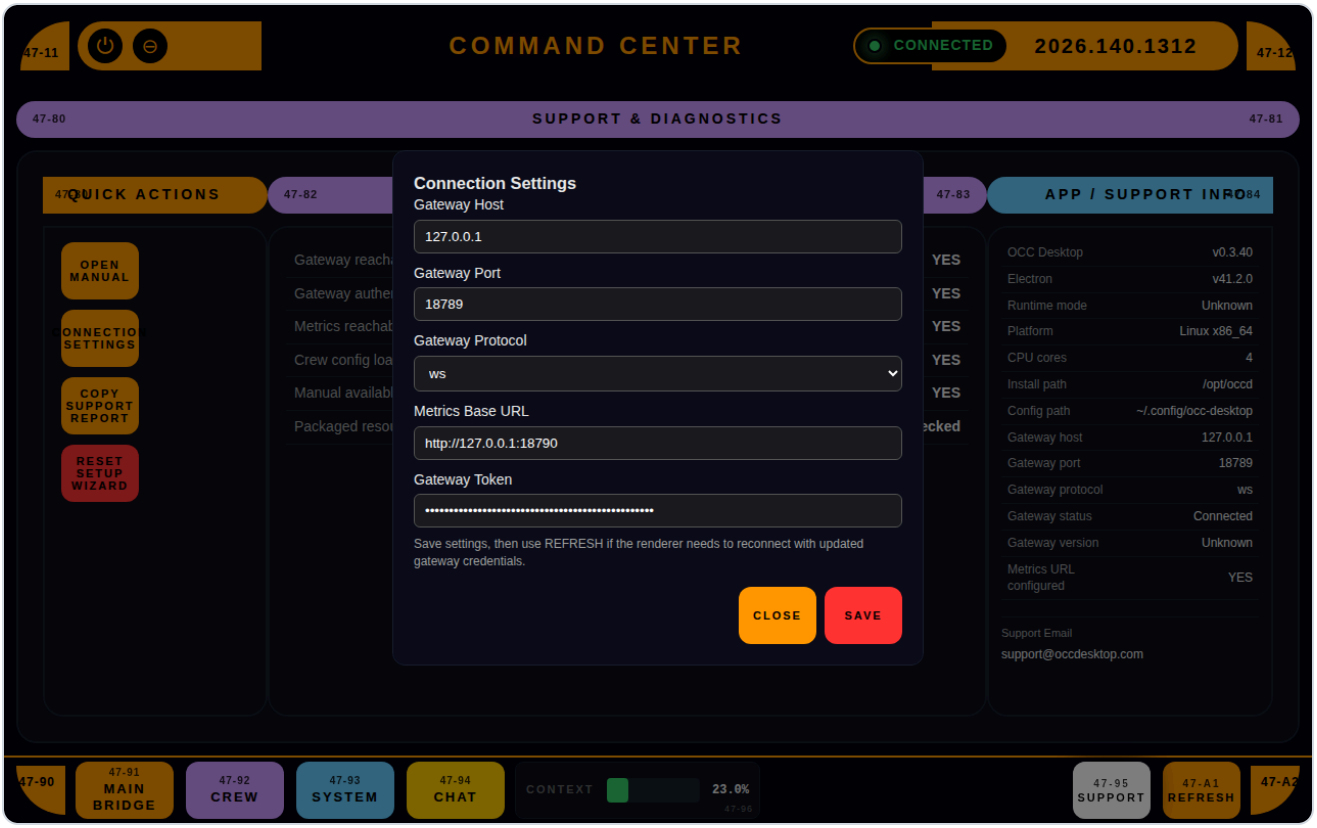
The center panel summarizes the current health of the connection and packaged application state. Use it to quickly confirm whether the gateway is reachable, whether OCC Desktop is authenticated, whether metrics are available, and whether key application resources are present.

Right Panel: App / Support Info

The right panel shows support-oriented reference data about the running application, including version details, install path, config path, gateway configuration, and environment information. This is useful when validating that the installed app matches the expected build.

Connection Settings from Support

The Support page also acts as the home for connection settings. Use it when you need to correct gateway values, update an expired token, or verify exactly how OCC Desktop is configured to reach your OpenClaw environment.



Connection settings can be opened directly from the Support view when gateway or token values need to be corrected.

Connection Settings Fields

Field	Description
Gateway Host	IP address or hostname of the OpenClaw gateway
Gateway Port	WebSocket port (default: `18789`)
Gateway Protocol	`ws` for local, `wss` for TLS-secured remote
Metrics Base URL	Full URL to the metrics server (for example `http://host:18790`)
Gateway Token	Your operator token — treat like a password

Token Security

- treat your gateway token like a password
- if you suspect it leaked, rotate it immediately and update OCC Desktop

- retrieve your current token with `python3 -c 'import json, pathlib; print(json.load(open(pathlib.Path.home()/".openclaw/openclaw.json"))["gateway"]["auth"]["token"])'`

REFRESH

Use **REFRESH** after:

- starting or restarting the gateway
- approving pairing requests
- changing crew definitions
- updating connection settings

Alerts

OCC Desktop may surface alerts for:

- connection loss
- invalid or unauthorized token
- metrics endpoint failures

When an alert appears:

1. read the exact message
2. verify gateway reachability
3. confirm token validity with `python3 -c 'import json, pathlib; print(json.load(open(pathlib.Path.home()/".openclaw/openclaw.json"))["gateway"]["auth"]["token"])'`
4. click **REFRESH**

Use the Support page when you need to:

- check connection and runtime status
- open connection settings
- copy a support report
- review support-oriented application information
- reset onboarding for a fresh setup flow
- gather information before troubleshooting or reporting a problem

10. Pairing Nodes

What is pairing?

Pairing is the trust and authorization step that allows a node or device to participate in your OpenClaw environment.

Pairing is relevant when:

- a new node tries to connect to the gateway
- you've reset a device
- you've changed gateway security policies

How pairing requests appear in OCC Desktop

When pairing is required, you'll see:

- a **PAIRING PENDING** badge in the header
- Activity Feed entries describing the pairing request
- in v0.3.50, a dedicated pairing interstitial after onboarding for new unapproved desktop devices

Approving Pairing Requests

For OCC Desktop v0.3.50, the normal workflow is:

1. complete onboarding
2. when the pairing screen appears, click **Open WebUI**
3. approve the pending **OCC Desktop** request in OpenClaw WebUI under `/nodes`
4. return to OCC Desktop and click **Already Approved**
5. if needed, click **REFRESH** after approval

If your environment supports CLI approval, you may also approve via:

```
openclaw pair approve <node-id>
```

Pro tip: Pairing problems are often stale token issues in disguise. If pairing requests appear but approvals don't stick, confirm your gateway token is current.

11. Troubleshooting

Gateway token expired or invalid

Symptom: OCC Desktop shows unauthorized errors or cannot connect after previously working.

Fix:

- retrieve a fresh token with `python3 -c 'import json, pathlib; print(json.load(open(pathlib.Path.home()/".openclaw/openclaw.json"))["gateway"]["auth"]["token"])'`
- update the token in Settings → Connection Settings → Save
- click **REFRESH**

Metrics Base URL unreachable (but gateway works)

Symptom: gateway connection works, crew and sessions appear, but gauges show no data or errors.

Fix:

- verify the metrics server is running on the gateway host (default port: 18790)
- confirm the Metrics Base URL in Settings is correct and includes `http://` or `https://`
- test the URL in a browser

ws vs wss protocol mismatch

Symptom: connection fails immediately with a WebSocket error.

Fix:

- use `ws` locally or on LAN without TLS
- use `wss` for remote internet connections with TLS
- ensure the gateway has a valid TLS certificate configured if using `wss`
- switch protocol in Settings and then click **REFRESH**

Model not allowed when spawning a subagent

Symptom: a crew member fails to spawn with an error indicating the model is blocked.

Cause: your OpenClaw configuration restricts allowed models.

Fix: add the required model to the allowed models list, or change the crew member to use an already allowed model.

Subagent not appearing in the crew panel

Symptom: a subagent is running but OCC Desktop doesn't map it to the expected crew member.

Most common cause: the subagent label or name doesn't match the configured crew name exactly.

Fix: ensure the spawn label equals the crew name in OCC Desktop exactly.

White screen on launch

Symptom: the app launches to a blank or white window.

Likely causes: corrupted build artifacts, missing runtime libraries, or GPU/driver quirks.

Fix: reinstall the app. If building from source, rebuild and repackage. Launch from terminal to inspect logs.

Gateway connection fails

Symptom: OCC Desktop cannot connect, or TEST GATEWAY fails.

Checklist:

- gateway host is reachable
- correct port (default: 18789)
- correct protocol (`ws` vs `wss`)
- token is valid
- gateway service is running

Crew disappears on refresh

Symptom: crew list clears or doesn't repopulate after refresh.

Status: fixed in modern OCC Desktop builds. If you still see it on an older version, upgrade.

12. Reference

Keyboard Shortcuts

Action	Shortcut
Refresh view	`Ctrl+R`
Hard reload	`Ctrl+Shift+R`
Toggle DevTools	`Ctrl+Shift+I`
Full Screen Toggle	`F11`
Quit	`Ctrl+Q`

> **Pro tip:** If something looks stuck, open DevTools and check the Console tab — gateway auth errors often appear there.

File Locations

Item	Location
Install directory	`/opt/occd/`
App configuration	`~/.config/occ-desktop/`
App cache	`~/.cache/occ-desktop/`
Uninstaller	`/opt/occd/maintenancetool`

Tech Stack

Component	Details
Desktop shell	Electron (Chromium + Node.js)
Frontend	React + TypeScript + Vite
UI style	Dark-themed mission control
Gateway protocol	WebSocket (`ws` / `wss`)
Packaging	Qt Installer Framework `.run` installer

End of manual — OCC Desktop v0.3.50