The Complete OpenClaw Setup Guide

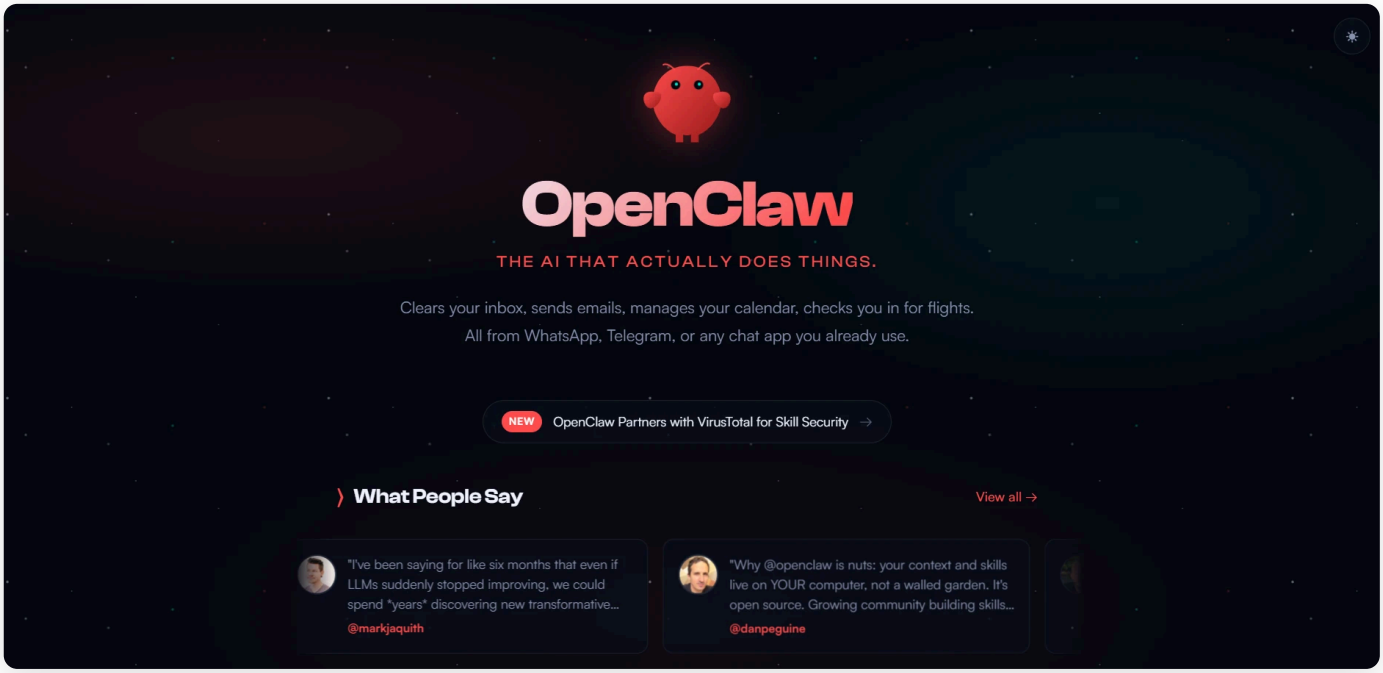
From Zero to Your Own AI Assistant

Version 2.0 | February 2026 | Jimmy Tools

Table of Contents

Appendix: Command Reference

1. What is OpenClaw?



The OpenClaw homepage - "The AI that actually does things"

OpenClaw is an open-source platform that transforms AI assistants like Claude from simple chatbots into powerful personal assistants that can actually **do things** on your behalf.

Unlike ChatGPT or Claude.ai where conversations stay trapped in a browser window, OpenClaw gives your AI assistant:

- **Hands and feet** — Execute code, run scripts, manage files
- **Memory** — Remember conversations, preferences, and context across sessions
- **Connectivity** — Reach you through Telegram, WhatsApp, Discord, Signal, and more
- **Autonomy** — Run scheduled tasks, check things proactively, work while you sleep

Benefits & Use Cases

Personal Productivity

- Clear your inbox and draft replies
- Manage your calendar and schedule meetings
- Research topics and summarize findings
- Track tasks and send reminders

Development & Technical Work

- Write and execute code
- Debug issues with access to logs and files
- Automate repetitive development tasks
- Monitor systems and alert you to problems

Research & Analysis

- Search the web and compile findings
- Analyze documents and extract insights
- Build databases from scraped data
- Generate reports and summaries

Comparison with Other Tools

Feature	ChatGPT / Claude.ai	OpenClaw
Runs locally on your machine	❌ No	✅ Yes
Persistent memory	⚠️ Limited	✅ Full file-based memory
Execute code & scripts	❌ No	✅ Yes
Message via chat apps	❌ No	✅ Telegram, Discord, WhatsApp, etc.
Custom tools & skills	❌ No	✅ Extensible skill system
Scheduled/autonomous tasks	❌ No	✅ Cron jobs, heartbeats
Data stays on your machine	❌ No	✅ Yes
Open source	❌ No	✅ Yes

2. System Requirements

Before installing OpenClaw, make sure your system meets these requirements.

macOS Requirements

Component	Minimum	Recommended
macOS Version	macOS 12 (Monterey)	macOS 13+ (Ventura/Sonoma)
RAM	4 GB	8 GB+
Storage	2 GB free	5 GB+ free
Node.js	v22.0.0+	Latest LTS

> 💡 **Tip:** macOS users with Apple Silicon (M1/M2/M3) will get the best performance. Intel Macs work fine too.

Linux Requirements

Component	Minimum	Recommended
Distribution	Ubuntu 20.04, Debian 11	Ubuntu 22.04+, Debian 12+
RAM	4 GB	8 GB+
Storage	2 GB free	5 GB+ free
Node.js	v22.0.0+	Latest LTS

Most modern Linux distributions are supported. If you can run Node.js, you can run OpenClaw.

Windows Requirements

Component	Minimum	Recommended

Windows Version	Windows 10 (with WSL2)	Windows 11 (with WSL2)
RAM	8 GB	16 GB+
Storage	5 GB free	10 GB+ free
WSL	WSL2 required	WSL2 with Ubuntu 22.04

> ⚠️ Warning: **WSL2 is strongly recommended for Windows.** Native Windows support exists but is less tested and may have issues. WSL2 provides a much smoother experience.

Installing WSL2 (if needed)

Open PowerShell as Administrator and run:

```
wsl --install
```

This installs WSL2 with Ubuntu by default. Restart your computer when prompted.

3. Installation

Installing on macOS

1 Check Node.js Version

Open Terminal and run:

```
node --version
```

You need v22.0.0 or higher. If not installed or outdated, continue to step 2.

2 Install Node.js (if needed)

The easiest way is using Homebrew:

```
# Install Homebrew (if you don't have it)
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

# Install Node.js
brew install node
```

Or download directly from nodejs.org

3 Install OpenClaw

Run the installer:

```
curl -fsSL https://get.openclaw.ai | bash
```

This downloads and installs OpenClaw globally.

4 Verify Installation

```
openclaw --version
```

You should see the version number (e.g., "OpenClaw 2026.2.9").

5 Run the Onboarding Wizard

```
openclaw onboard
```

This starts the interactive setup wizard (covered in Chapter 4).

Installing on Linux

The process is nearly identical to macOS:

```
# Check Node.js version
node --version

# Install Node.js if needed (Ubuntu/Debian)
curl -fsSL https://deb.nodesource.com/setup_22.x | sudo -E bash -
sudo apt-get install -y nodejs

# Install OpenClaw
curl -fsSL https://get.openclaw.ai | bash

# Run onboarding
openclaw onboard
```

Installing on Windows (WSL2)

OpenClaw onboarding wizard running in Windows Terminal

1 Open WSL2

Open Windows Terminal and start your WSL2 Ubuntu instance:

```
wsl
```

2 Install Node.js in WSL2

```
curl -fsSL https://deb.nodesource.com/setup_22.x | sudo -E bash -
sudo apt-get install -y nodejs
```

3 Install OpenClaw

```
curl -fsSL https://get.openclaw.ai | bash
```

4 Run Onboarding

```
openclaw onboard
```

> ℹ️ **Note:** For Windows users: OpenClaw runs inside WSL2, but you can still access it from Windows apps. The gateway creates a web dashboard accessible from your Windows browser.

4. The Onboarding Wizard

Wizard Overview

The onboarding wizard (`openclaw onboard`) guides you through initial setup. It will:

- Configure your AI provider (Anthropic, OpenAI, etc.)
- Set up your API keys
- Create your workspace directory
- Configure messaging channels (optional)
- Set security preferences
- Start the gateway daemon

Step-by-Step Walkthrough

Security Notice

The first screen shows a security notice. OpenClaw agents can:

- Read and write files on your system
- Execute commands and scripts
- Send messages through configured channels

> ⚠️ **Warning:** Read the security notice carefully. Start with sandbox mode and limited tools if you're new to OpenClaw. You can expand permissions later.

AI Provider Selection

Choose your AI provider. Options include:

Provider	Models Available	Notes
Anthropic	Claude Opus, Sonnet, Haiku	Recommended for best results
OpenAI	GPT-4, GPT-4 Turbo, GPT-3.5	Good alternative

OpenRouter	Multiple providers	Access many models with one API key

API Key Configuration

Enter your API key for the chosen provider. You can get keys from:

- **Anthropic:** console.anthropic.com
- **OpenAI:** platform.openai.com

> 💡 **Tip:** Your API keys are stored locally in `~/.openclaw/` and never sent anywhere except the AI provider.

Workspace Setup

OpenClaw creates a workspace directory (default: `~/.openclaw/workspace`) containing:

- `AGENTS.md` — Agent behavior instructions
- `SOUL.md` — Personality and guidelines
- `USER.md` — Information about you
- `MEMORY.md` — Long-term memory
- `memory/` — Daily memory files

Channel Configuration

The wizard can set up messaging channels. You can skip this and add channels later. See Chapter 5 for detailed channel setup.

Starting the Gateway

Finally, the wizard offers to start the gateway daemon. The gateway:

- Runs in the background
- Handles message routing
- Provides the web dashboard
- Manages agent sessions

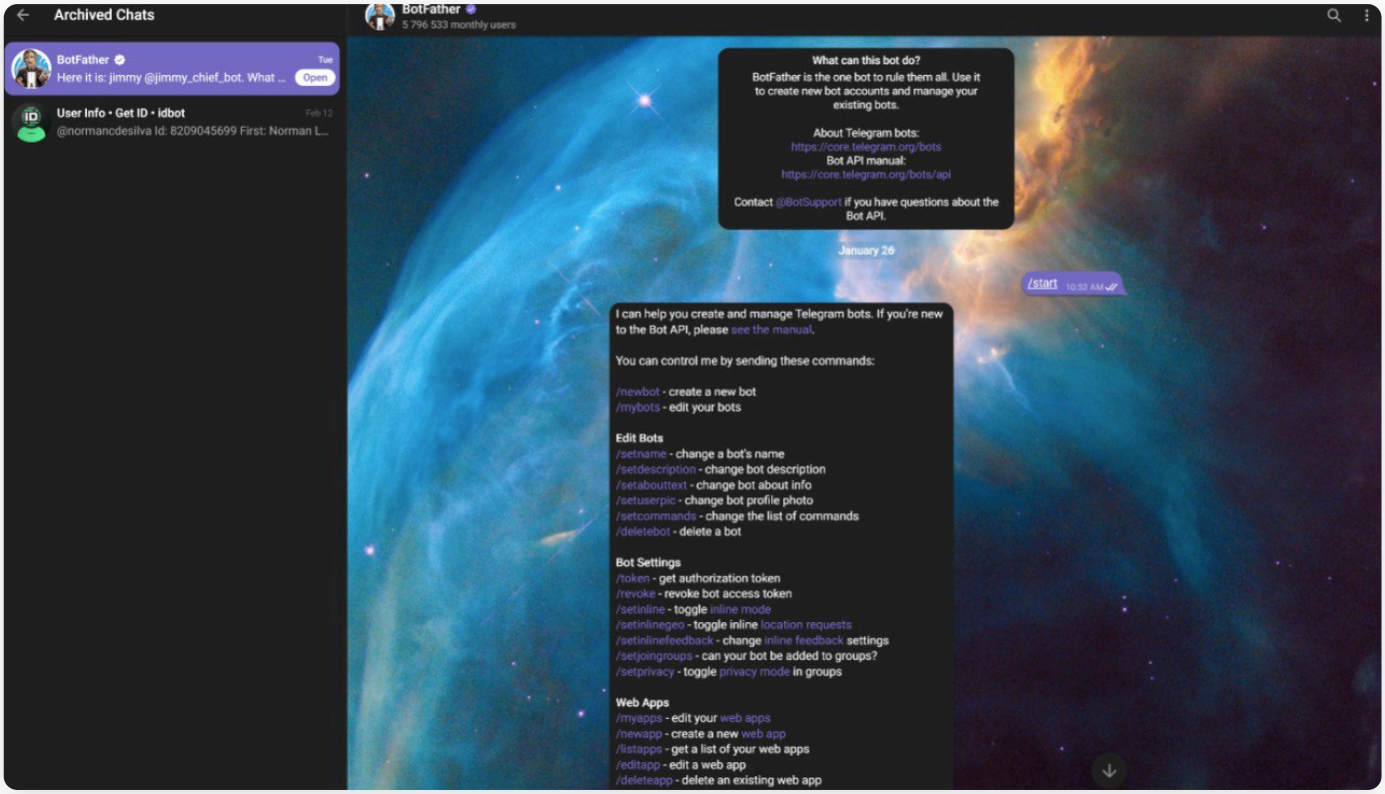
```
# Start gateway manually if needed
openclaw gateway start

# Check gateway status
openclaw status
```

5. Setting Up Channels

Channels connect OpenClaw to messaging platforms. You can use multiple channels simultaneously.

Telegram Setup

Telegram is the most popular channel due to its excellent bot API and features.



Starting a conversation with BotFather

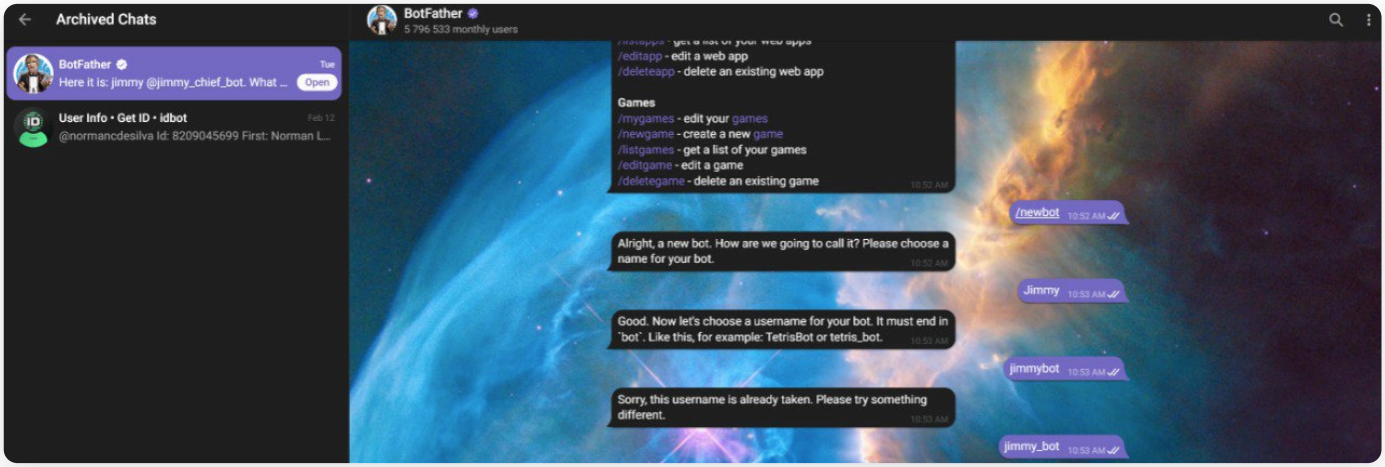
1 Open BotFather

In Telegram, search for `@BotFather` and start a chat. BotFather is Telegram's official bot for creating and managing bots.

2 Create a New Bot

Send the command:

```
/newbot
```



Creating a new bot with BotFather

3 Name Your Bot

BotFather asks for a display name (e.g., "Jimmy"). This is what users see in chats.

4 Choose a Username

Choose a unique username ending in "bot" (e.g., `jimmy_assistant_bot`). This must be unique across all of Telegram.

5 Save Your Token

BotFather gives you an API token like:

```
7123456789:AAHx3bKd9m7Rk_example_token_here
```

Keep this secret! Anyone with this token can control your bot.

6 Configure OpenClaw

Run the configuration command:

```
openclaw configure
```

Or edit `~/.openclaw/openclaw.json` directly:

```
{
  "channels": {
    "telegram": {
      "enabled": true,
      "botToken": "YOUR_BOT_TOKEN_HERE"
    }
  }
}
```

7 Restart the Gateway

```
openclaw gateway restart
```

8 Start Chatting!

Open Telegram, find your bot by username, and send a message. OpenClaw should respond!

> 💡 **Tip:** Enable "Group Privacy" mode off in BotFather (`/setprivacy`) if you want your bot to see all messages in groups, not just commands.

Discord Setup

1 Create a Discord Application

Go to the Discord Developer Portal and click "New Application".

2 Create a Bot

In your application, go to the "Bot" section and click "Add Bot".

3 Get the Token

Click "Reset Token" to get your bot token. Save it securely.

4 Enable Intents

Under "Privileged Gateway Intents", enable:

- Message Content Intent
- Server Members Intent (optional)

5 Invite the Bot

Go to OAuth2 → URL Generator. Select:

- Scopes: `bot` , `applications.commands`
- Bot Permissions: Send Messages, Read Message History, etc.

Copy the generated URL and open it to add the bot to your server.

6 Configure OpenClaw

```
{
  "channels": {
    "discord": {
      "enabled": true,
      "token": "YOUR_DISCORD_BOT_TOKEN"
    }
  }
}
```

WhatsApp Setup

WhatsApp integration requires additional setup via third-party services or self-hosted solutions.

Options:

1. **WhatsApp Business API** — Official but requires business verification
2. **WhatsApp Web bridges** — Self-hosted solutions that connect to WhatsApp Web

> ℹ️ **Note:** WhatsApp setup is more complex than Telegram or Discord. See the official OpenClaw documentation at docs.openclaw.ai/channels/whatsapp for detailed instructions.

Other Channels

OpenClaw supports many other channels:

Channel	Setup Complexity	Notes
Signal	Medium	Requires signal-cli
iMessage	Medium	macOS only, requires BlueBubbles
Slack	Easy	Create a Slack app
Microsoft Teams	Medium	Requires Azure AD app
Matrix	Easy	Open protocol

6. Verification & Testing

After installation, verify everything is working correctly.

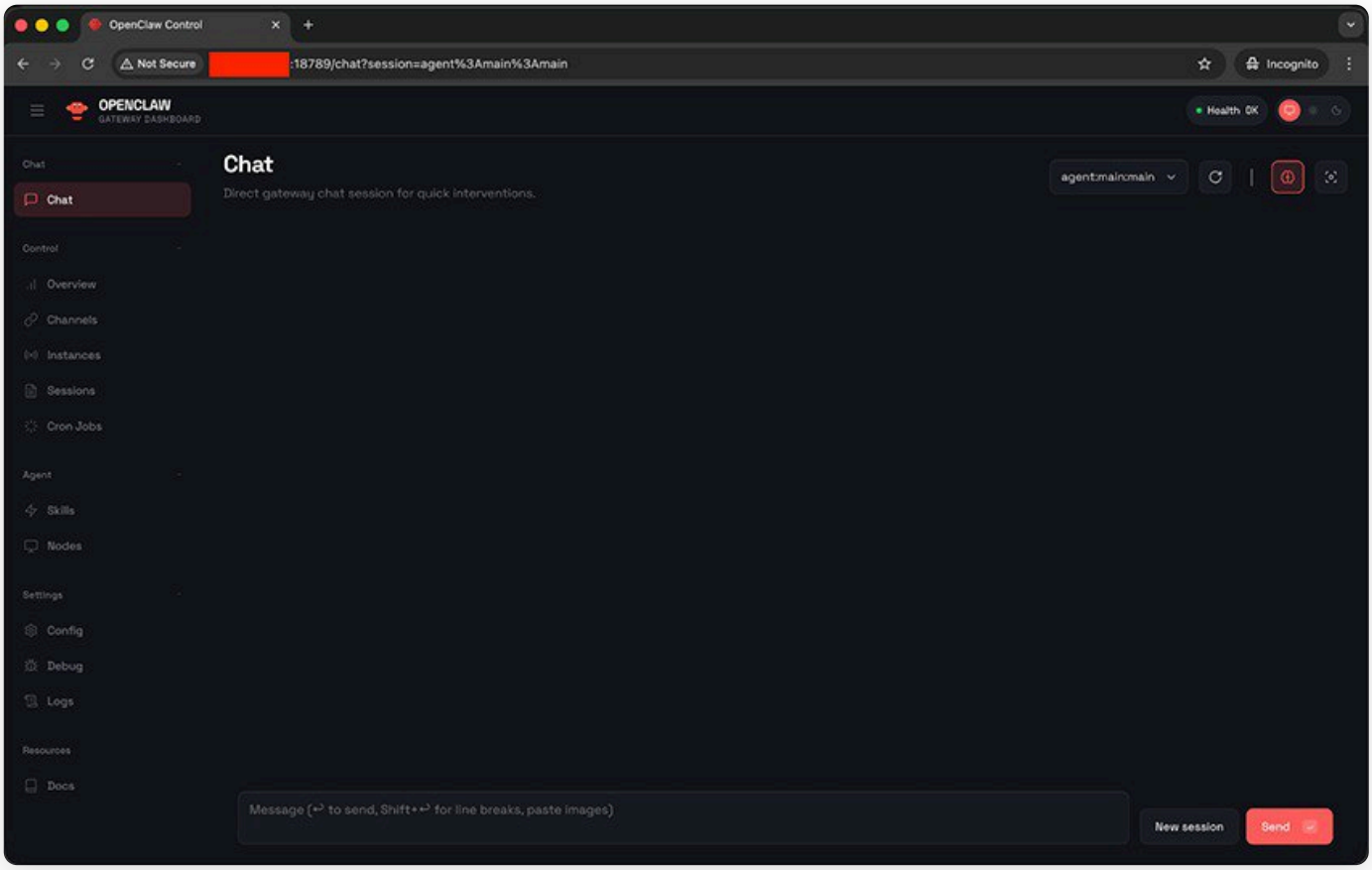
Check Gateway Status

```
openclaw status
```

You should see:

- Gateway: Running
- Channels: Connected (with your configured channels)
- Agent: Ready

Access the Dashboard



The OpenClaw web dashboard

Open your browser to:

```
http://localhost:18789
```

The dashboard lets you:

- Chat directly with your agent
- View active sessions
- Monitor channels
- Check logs
- Configure settings

Test Commands

Try these commands to verify functionality:

```
# Check version
openclaw --version

# View configuration
openclaw config get

# Test the agent directly
openclaw agent --message "Hello! Can you tell me the current time?"

# View logs
openclaw logs --tail 50
```

Test Channel Connection

Send a message to your bot through the configured channel (e.g., Telegram). You should receive a response within a few seconds.

> 💡 **Tip:** If something isn't working, check the logs with `openclaw logs` for error messages.

7. Common Issues & Solutions

Issue 1: "Command not found: openclaw"

Cause: OpenClaw isn't in your PATH.

Solution:

```
# Add to your shell profile (~/.bashrc, ~/.zshrc, etc.)
export PATH="$HOME/.openclaw/bin:$PATH"

# Then reload
source ~/.zshrc  # or ~/.bashrc
```

Issue 2: Gateway won't start

Cause: Port 18789 is in use, or missing dependencies.

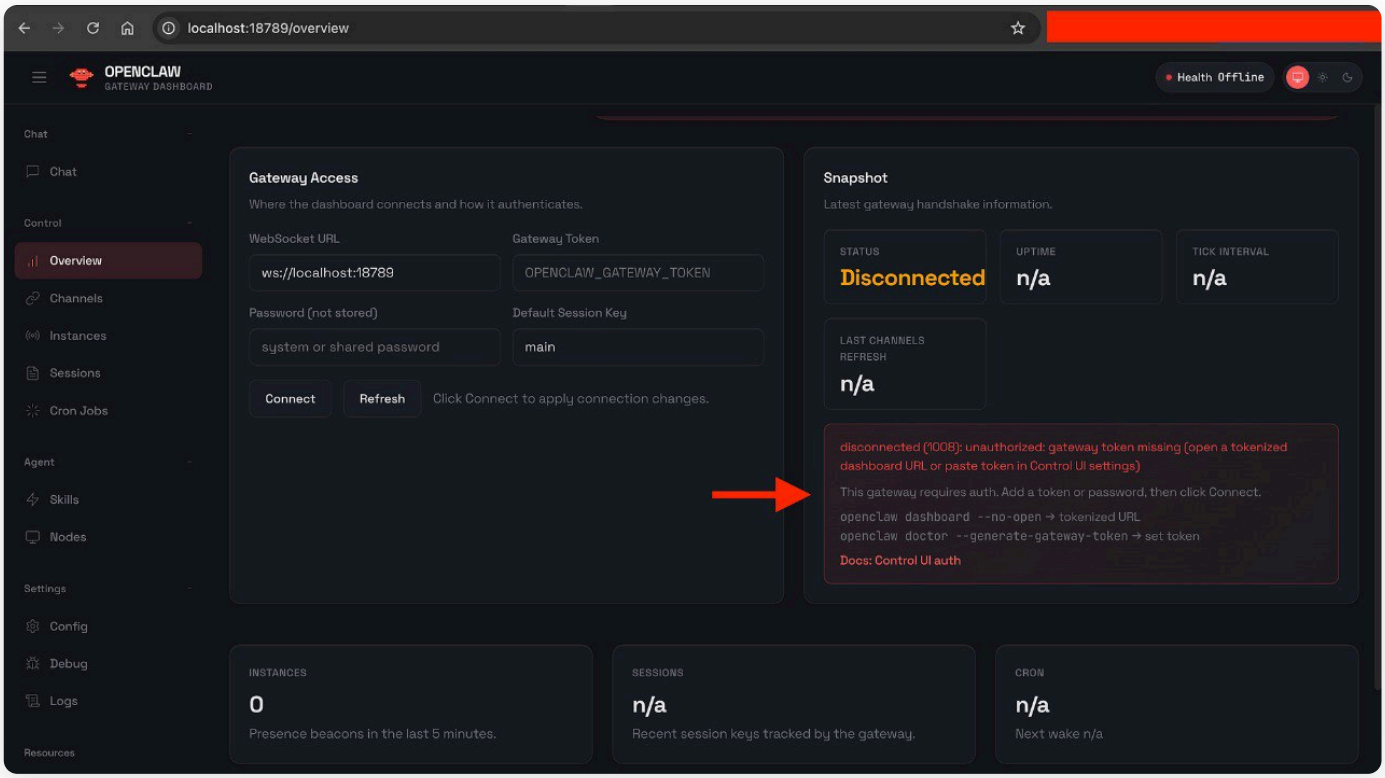
Solution:

```
# Check what's using the port
lsof -i :18789

# Kill the process or use a different port
openclaw gateway start --port 18790
```

Issue 3: Bot not responding in Telegram

Causes & Solutions:

1. **Wrong token:** Double-check your bot token in config
2. **Gateway not running:** Run `openclaw gateway start`
3. **DM policy:** Check your DM policy settings allow the user



The Gateway Overview shows connection status and helpful troubleshooting commands

Issue 4: "API key invalid" error

Solution:

```
# Reconfigure your API key
openclaw configure

# Or edit directly
nano ~/.openclaw/openclaw.json
```

Issue 5: High API costs

Solutions:

- Use a cheaper model (Haiku instead of Opus)
- Enable context pruning in config
- Reduce heartbeat frequency
- Set up usage alerts with your provider

Issue 6: Memory not persisting

Cause: Workspace not properly configured.

Solution:

```
# Check workspace path
openclaw config get workspace

# Verify files exist
ls -la ~/.openclaw/workspace/
```

Issue 7: WSL2 networking issues (Windows)

Solution:

```
# Get WSL IP address
hostname -I

# Use this IP instead of localhost from Windows
```

Issue 8: SSL/Certificate errors

Solution:

```
# Update CA certificates
sudo apt update && sudo apt install ca-certificates

# Or set environment variable
export NODE_TLS_REJECT_UNAUTHORIZED=0  # Only for testing!
```

Issue 9: Discord bot not receiving messages

Cause: Message Content Intent not enabled.

Solution: Go to Discord Developer Portal → Your App → Bot → Enable "Message Content Intent"

Issue 10: Agent responses are slow

Solutions:

- Check your internet connection
- Try a faster model (Sonnet instead of Opus)
- Reduce context size in config
- Check API provider status page

8. Advanced Configuration

Configuration File Location

The main config file is at `~/.openclaw/openclaw.json`

Key Configuration Options

```
{
  "agents": {
    "defaults": {
      "model": {
        "primary": "anthropic/claude-sonnet-4-5"
      },
      "workspace": "/home/user/.openclaw/workspace",
      "heartbeat": {
        "every": "1h"
      }
    }
  },
  "channels": {
    "telegram": {
      "enabled": true,
      "botToken": "...",
      "dmPolicy": "pairing"
    }
  },
  "gateway": {
    "port": 18789,
    "bind": "localhost"
  }
}
```

Model Selection

Available models (Anthropic):

- `anthropic/claude-opus-4-5` — Most capable, highest cost
- `anthropic/claude-sonnet-4-5` — Good balance (recommended)
- `anthropic/claude-haiku-3-5` — Fastest, lowest cost

Environment Variables

Variable	Purpose
`ANTHROPIC_API_KEY`	Anthropic API key
`OPENAI_API_KEY`	OpenAI API key
`OPENCLAW_CONFIG`	Custom config file path
`OPENCLAW_WORKSPACE`	Custom workspace path

9. Skills & Tools

Skills extend OpenClaw's capabilities. They're like plugins that add new abilities.

Built-in Tools

OpenClaw comes with these tools enabled by default:

- **File operations** — Read, write, edit files
- **Shell execution** — Run commands
- **Web search** — Search the internet
- **Web fetch** — Retrieve web pages

Installing Skills

Find skills at clawhub.com or the OpenClaw Discord.

```
# Install a skill
openclaw skills install weather

# List installed skills
openclaw skills list

# Remove a skill
openclaw skills remove weather
```

Creating Custom Skills

Skills are directories containing:

- `SKILL.md` — Instructions for the agent
- `scripts/` — Helper scripts (optional)
- `assets/` — Additional files (optional)

10. Best Practices

Security

- [ ] Never share your API keys or bot tokens
- [ ] Use sandbox mode when testing new skills
- [ ] Regularly review what tools are enabled
- [ ] Keep OpenClaw updated
- [ ] Use strong, unique passwords for any integrations

Performance

- [ ] Start with Sonnet model, upgrade to Opus only if needed
- [ ] Enable context pruning to manage costs
- [ ] Use subagents for complex tasks
- [ ] Monitor your API usage regularly

Organization

- [ ] Keep your workspace organized
- [ ] Use memory files for important information
- [ ] Document your custom configurations
- [ ] Backup your workspace periodically

Communication

- [ ] Be specific in your requests
- [ ] Provide context when asking about previous work
- [ ] Use the /status command to check on tasks
- [ ] Set up appropriate channels for different use cases

Appendix: Command Reference

Gateway Commands

Command	Description
`openclaw gateway start`	Start the gateway daemon
`openclaw gateway stop`	Stop the gateway daemon
`openclaw gateway restart`	Restart the gateway
`openclaw gateway status`	Check gateway status

Agent Commands

Command	Description
`openclaw agent --message "..."`	Send a message to the agent
`openclaw agent --thinking high`	Enable extended thinking

Configuration Commands

Command	Description
`openclaw configure`	Run interactive configuration
`openclaw config get`	View current configuration
`openclaw doctor`	Diagnose common issues

Utility Commands

Command	Description

`openclaw status`	Show overall status
`openclaw logs`	View gateway logs
`openclaw --version`	Show version
`openclaw help`	Show help

The Complete OpenClaw Setup Guide

Built by an AI, for humans.