Working with OpenClaw after Setup

You finished the setup guide. OpenClaw is running and you've already connected your first channel.

Most day-to-day tasks happen in the **dashboard** (a browser-based admin panel) or by **talking to the bot** in chat. You rarely need the terminal after setup.

---

1. Open the dashboard

The dashboard runs on your server. Since your server has no public ports, you reach it through an SSH tunnel.

Step 1. On your PC, open a terminal and create the tunnel:

```
ssh -L 18789:127.0.0.1:18789 remote@TS_IP
```

Keep this window open.

Step 2. In the same terminal (or another SSH session to the server), run:

```
openclaw dashboard --no-open
```

It prints a link like `http://127.0.0.1:18789/?token=...`

Step 3. Open that link in your PC's browser.

First-time device approval

The first time you open the dashboard from a new browser, you need to approve it from the terminal:

```
openclaw devices list
openclaw devices approve <requestId>
```

After approval, that browser is trusted and you won't need to do this again.

What you can do in the dashboard

Section	Purpose
Chat	Talk to the bot directly, see tool calls live
Channels	Manage Telegram, WhatsApp, Slack connections
Configuration	Edit settings with validation
Skills	Enable/disable skills, set API keys
Cron	View and manage scheduled tasks
Logs	Tail gateway logs with filtering

---

2. Approve new users

When someone messages the bot for the first time, they receive a pairing code. This is the one thing you must do from the terminal:

```
openclaw pairing approve <channel> <code>
```

For example, if someone on Telegram gets code `AB12CD`:

```
openclaw pairing approve telegram AB12CD
```

Pairing codes expire after 1 hour. Replace `telegram` with `whatsapp` or `slack` depending on where the person messaged.

---

3. Teach the bot

The easiest way to teach the bot is to just tell it. Send a message like:

> "Remember that my name is Alex and I prefer metric units."
>
> "Remember that our team standup is at 9 AM every weekday."
>
> "Forget what I told you about my favorite color."

The bot stores memories as plain files on disk. You don't need to manage these files — the bot reads and writes them on its own.

You can also talk to the bot from the dashboard's **Chat** panel, which is handy when you want to teach it something without going through Telegram or WhatsApp.

From the terminal (advanced)
Memory files live in the bot's workspace (`~/.openclaw/workspace`): - `MEMORY.md` — long-term facts and preferences - `memory/YYYY-MM-DD.md` — daily conversation notes You can edit these files directly if you prefer.

4. Customize personality

The quickest way to change the bot's personality is to ask it:

> "I want you to be more formal and concise."
>
> "Write a SOUL.md that makes you a cooking expert who never gives medical advice."
>
> "Rewrite your personality to be a friendly assistant named Max who speaks casually and always asks a follow-up question."

The bot updates its own personality file and the changes take effect on the next conversation. Send `/reset` in chat to start a fresh conversation with the new personality.

From the terminal (advanced)
The personality file is `~/.openclaw/workspace/SOUL.md`. You can edit it directly:

```
You are a friendly assistant named Max. You speak casually, use short sentenc
and always ask a follow-up question. You are knowledgeable about cooking and
gardening. You never give medical advice.
```

There is also `~/.openclaw/workspace/AGENTS.md` for agent identity and profile context, which is injected into the system prompt automatically.

---

5. Manage settings

Open the dashboard and go to the **Configuration** panel. It shows all settings in a visual editor and validates your changes before applying them.

Things you can change here:

- AI model and fallbacks
- Session reset policies (e.g., reset conversations daily)
- Message prefixes and reactions
- Channel allowlists and group policies
- Tool restrictions

The dashboard restarts the bot automatically when you save changes.

From the terminal

```
# Change a setting
openclaw config set <key.path> <value>
```

```
# Remove a setting
openclaw config unset <key.path>
```

Examples:

```
# Change the default model
openclaw config set agents.defaults.model.primary "claude-sonnet-4-5-20250929

# Reset conversations daily
openclaw config set session.reset.mode "daily"

# Disable Slack
openclaw config set channels.slack.enabled false
```

After manual edits, restart the gateway:

```
systemctl --user restart openclaw-gateway.service
```

6. Add skills

Open the dashboard and go to the **Skills** panel. You can enable or disable skills and enter API keys right there.

To browse skills that aren't installed yet, visit clawhub.com.

From the terminal

```
# Install a skill from ClawHub
clawhub install <skill-name>

# Disable a skill
openclaw config set skills.entries.<skill-name>.enabled false
```

Skills install to your workspace's `skills` folder. Treat third-party skills as untrusted code — read the skill's `SKILL.md` before enabling it.

7. Schedule tasks (cron)

You can have the bot do things on a schedule — send a daily summary, check something every hour, or fire a one-time reminder.

The easiest way is to ask the bot:

> "Every morning at 9 AM, summarize my unread emails and send it to Slack."
>
> "Remind me in 2 hours to check the deployment."
>
> "Every Monday at 8 AM, give me a weekly project status update."

The bot creates cron jobs that survive restarts. You can view and manage them in the dashboard's **Cron** panel — enable, disable, run manually, or check execution history.

From the terminal

```
# Add a recurring job (cron expression, 9 AM daily, Pacific time)
openclaw cron add --name "Morning summary" --cron "0 9 * * *" \
  --tz "America/Los_Angeles" --session isolated \
  --message "Summarize my unread emails" --announce --channel slack --to "cha

# Add a one-shot reminder
openclaw cron add --name "Check deploy" --at "2026-02-08T18:00:00Z" \
  --session isolated --message "Remind me to check the deployment"

# List all jobs
openclaw cron list

# Run a job manually
```

```
openclaw cron run <jobId>

# View run history
openclaw cron runs --id <jobId>

# Remove a job
openclaw cron remove <jobId>
```

Jobs are stored in `~/.openclaw/cron/jobs.json` and persist across restarts. One-shot jobs auto-delete after they run.

---

8. Multiple agents (advanced)

You can run multiple bot personalities, each with its own memory, workspace, and channel bindings. This is for users who want two different bots — for example, a casual assistant on Telegram and a research bot on Slack.

This section requires the terminal.

Step 1. Create a workspace

```
mkdir -p ~/.openclaw/workspace-helper
```

Step 2. Give it a personality

Create a `SOUL.md` in the new workspace (or ask your existing bot to help you write one and copy it over):

```
You are a concise research assistant. You give short, factual answers with sour
You never use emojis or casual language. When unsure, you say so.
```

Step 3. Edit the configuration

In the dashboard **Configuration** panel (or by editing `~/.openclaw/openclaw.json`), add the new agent and a binding:

```
{
  agents: {
    list: [
      {
        id: "default",
        workspace: "~/.openclaw/workspace"
      },
      {
        id: "helper",
        workspace: "~/.openclaw/workspace-helper"
      }
    ]
  },
  bindings: [
    { agentId: "default", match: { channel: "telegram" } },
    { agentId: "helper", match: { channel: "slack" } }
  ]
}
```

Bindings use specificity-based matching (highest to lowest priority):

1. Exact peer (specific DM or group ID)
2. Account ID
3. Channel-wide match
4. Fallback to default agent

Step 4. Restart

```
systemctl --user restart openclaw-gateway.service
```

Each agent now has isolated sessions, memory, and personality.

9. Keep it running

These are maintenance tasks you'll do occasionally, not daily.

Update OpenClaw

```
pnpm update -g openclaw
openclaw doctor
systemctl --user restart openclaw-gateway.service
```

The `doctor` command validates your config and runs any needed migrations.

Health check

```
openclaw doctor
```

Run this after updates or if something seems off.

Restart the gateway

```
systemctl --user restart openclaw-gateway.service
```

View logs

Use the dashboard **Logs** panel for filtered, searchable logs.

From the terminal

```
journalctl --user -u openclaw-gateway.service -n 100 --no-pager
```

10. Quick reference

Commands you'll use after updates:

```
pnpm update -g openclaw        # Update OpenClaw
openclaw doctor                 # Validate config, run migrations
systemctl --user restart openclaw-gateway.service  # Restart
```

Commands you'll rarely need:

```
openclaw dashboard --no-open    # Start the dashboard
openclaw devices list           # List pending device requests
openclaw devices approve <id>   # Approve a new browser
openclaw pairing approve <channel> <code>  # Approve a new user
openclaw config set <key> <value>          # Change a setting
openclaw cron list              # List scheduled jobs
openclaw cron run <jobId>       # Run a job manually
openclaw security audit --deep  # Security audit
```