Other install methods

Podman

Run the OpenClaw gateway in a **rootless** Podman container. Uses the same image as Docker (build from the repo **Dockerfile**).

Requirements

- Podman (rootless)
- Sudo for one-time setup (create user, build image)

Quick start

1. One-time setup (from repo root; creates user, builds image, installs launch script):

```
./setup-podman.sh
```

This also creates a minimal `~openclaw/.openclaw/openclaw.json` (sets `gateway.mode="local"`) so the gateway can start without running the wizard.

By default the container is **not** installed as a systemd service, you start it manually (see below). For a production-style setup with auto-start and restarts, install it as a systemd Quadlet user service instead:

```
./setup-podman.sh --quadlet
```

(Or set `OPENCLAW_PODMAN_QUADLET=1`; use `--container` to install only the container and launch script.)

2. Start gateway (manual, for quick smoke testing):

```
./scripts/run-openclaw-podman.sh launch
```

3. Onboarding wizard (e.g. to add channels or providers):

```
./scripts/run-openclaw-podman.sh launch setup
```

Then open `http://127.0.0.1:18789/` and use the token from `~openclaw/.openclaw/.env` (or the value printed by setup).

Systemd (Quadlet, optional)

If you ran `./setup-podman.sh --quadlet` (or `OPENCLAW_PODMAN_QUADLET=1`), a unit is installed so the gateway runs as a systemd user service for the openclaw user. The service is enabled and started at the end of setup.

- **Start:** `sudo systemctl --machine openclaw@ --user start openclaw.service`
- **Stop:** `sudo systemctl --machine openclaw@ --user stop openclaw.service`
- **Status:** `sudo systemctl --machine openclaw@ --user status openclaw.service`
- **Logs:** `sudo journalctl --machine openclaw@ --user -u openclaw.service -f`

The quadlet file lives at `~openclaw/.config/containers/systemd/openclaw.container`. To change ports or env, edit that file (or the `.env` it sources), then `sudo systemctl --

`machine openclaw@ --user daemon-reload` and restart the service. On boot, the service starts automatically if lingering is enabled for openclaw (setup does this when loginctl is available).

To add quadlet **after** an initial setup that did not use it, re-run: `./setup-podman.sh --quadlet .`

The openclaw user (non-login)

`setup-podman.sh` creates a dedicated system user `openclaw`:

Shell: `nologin` — no interactive login; reduces attack surface.

Home: e.g. `/home/openclaw` — holds `~/.openclaw` (config, workspace) and the launch script `run-openclaw-podman.sh`.

Rootless Podman: The user must have a **subuid** and **subgid** range. Many distros assign these automatically when the user is created. If setup prints a warning, add lines to `/etc/subuid` and `/etc/subgid`:

```
openclaw:100000:65536
```

Then start the gateway as that user (e.g. from cron or systemd):

```
sudo -u openclaw /home/openclaw/run-openclaw-podman.sh
sudo -u openclaw /home/openclaw/run-openclaw-podman.sh setup
```

Config: Only `openclaw` and root can access `/home/openclaw/.openclaw`. To edit config: use the Control UI once the gateway is running, or `sudo -u openclaw $EDITOR /home/openclaw/.openclaw/openclaw.json`.

Environment and config

Token: Stored in `~openclaw/.openclaw/.env` as `OPENCLAW_GATEWAY_TOKEN`. `setup-podman.sh` and `run-openclaw-podman.sh` generate it if missing (uses `openssl`, `python3`, or `od`).

Optional: In that `.env` you can set provider keys (e.g. `GROQ_API_KEY`, `OLLAMA_API_KEY`) and other OpenClaw env vars.

Host ports: By default the script maps `18789` (gateway) and `18790` (bridge). Override the **host** port mapping with `OPENCLAW_PODMAN_GATEWAY_HOST_PORT` and `OPENCLAW_PODMAN_BRIDGE_HOST_PORT` when launching.

Paths: Host config and workspace default to `~openclaw/.openclaw` and `~openclaw/.openclaw/workspace`. Override the host paths used by the launch script with `OPENCLAW_CONFIG_DIR` and `OPENCLAW_WORKSPACE_DIR`.

Useful commands

Logs: With quadlet: `sudo journalctl --machine openclaw@ --user -u openclaw.service -f`. With script: `sudo -u openclaw podman logs -f openclaw`

Stop: With quadlet: `sudo systemctl --machine openclaw@ --user stop openclaw.service`. With script: `sudo -u openclaw podman stop openclaw`

Start again: With quadlet: `sudo systemctl --machine openclaw@ --user start openclaw.service`. With script: re-run the launch script or `podman start openclaw`

Remove container: `sudo -u openclaw podman rm -f openclaw` — config and workspace on the host are kept

Troubleshooting

Permission denied (EACCES) on config or auth-profiles: The container defaults to `--userns=keep-id` and runs as the same uid/gid as the host user running the script. Ensure your host `OPENCLAW_CONFIG_DIR` and `OPENCLAW_WORKSPACE_DIR` are owned by that user.

- **Gateway start blocked (missing `gateway.mode=local`):** Ensure `~openclaw/.openclaw/openclaw.json` exists and sets `gateway.mode="local"`. `setup-podman.sh` creates this file if missing.
- **Rootless Podman fails for user openclaw:** Check `/etc/subuid` and `/etc/subgid` contain a line for `openclaw` (e.g. `openclaw:100000:65536`). Add it if missing and restart.
- **Container name in use:** The launch script uses `podman run --replace`, so the existing container is replaced when you start again. To clean up manually: `podman rm -f openclaw`.
- **Script not found when running as openclaw:** Ensure `setup-podman.sh` was run so that `run-openclaw-podman.sh` is copied to openclaw's home (e.g. `/home/openclaw/run-openclaw-podman.sh`).
- **Quadlet service not found or fails to start:** Run `sudo systemctl --machine openclaw@ --user daemon-reload` after editing the `.container` file. Quadlet requires cgroups v2: `podman info --format '{{.Host.CgroupsVersion}}'` should show `2`.

Optional: run as your own user

To run the gateway as your normal user (no dedicated openclaw user): build the image, create `~/.openclaw/.env` with `OPENCLAW_GATEWAY_TOKEN`, and run the container with `--userns=keep-id` and mounts to your `~/.openclaw`. The launch script is designed for the openclaw-user flow; for a single-user setup you can instead run the `podman run` command from the script manually, pointing config and workspace to your home. Recommended for most users: use `setup-podman.sh` and run as the openclaw user so config and process are isolated.

‹ Docker Nix ›