Clawdbot setup (Proxmox)

Goal

Run **Clawdbot** (long-running, local-first AI agent) reliably on **Proxmox**, mainly inside an **unprivileged LXC**, treating it like a server app, not a one-off chatbot.

Key Concepts

- Clawdbot is a Node.js agent that:
  - Executes shell commands, reads/writes files, keeps memory, connects to Telegram/WhatsApp, and runs continuously.
- Local-first: all config and memory live on disk under the user's home (`~/.clawdbot`), so you need persistent storage and backups.
- Proxmox options:
  - **VM**: easiest (full systemd, user sessions) but heavier.
  - **LXC**: lightweight and ideal for services, but no real user session, `systemd --user`, or user D-Bus, so the normal Clawdbot daemon installer fails.

Requirements

- Hardware:
  - Cloud LLM: 2 vCPU, 2–4 GB RAM.
  - Local models: 16+ GB RAM, GPU passthrough recommended.
  - Storage: at least 20 GB.
- OS: Prefer **Debian 12**, Ubuntu 22.04/24.04 also OK.
- Node.js: **22+ required**; install from NodeSource, not distro apt.

Strategy A: LXC + System-Level Service

1. Create LXC

- Debian 12 template.
- Unprivileged container.
- Static IP recommended.

2. Install deps + Node.js 22

Inside the container (as root):

```
apt update && apt upgrade -y
apt install -y curl git build-essential sudo python3

curl -fsSL https://deb.nodesource.com/setup_22.x -o nodesource_setup.sh
```

```
bash nodesource_setup.sh
apt install -y nodejs
node -v  # should show v22.x.x
```

3. Create non-root user

```
adduser clawd
usermod -aG sudo clawd
su - clawd
```

4. Install Clawdbot

```
npm install -g clawdbot@latest
```

> Do **not** run `clawdbot onboard --install-daemon` in LXC; `systemd --user` and user D-Bus are missing, so it will fail.

5. Create systemd service (as root)

```
sudo nano /etc/systemd/system/clawdbot.service
```

Paste:

```
[Unit]
Description=Clawdbot Gateway Service
Documentation=https://github.com/clawdbot/clawdbot
After=network.target

[Service]
User=clawd
Group=clawd
Environment=HOME=/home/clawd
Environment=NODE_ENV=production
Environment=CLAWDBOT_GATEWAY_BIND=0.0.0.0
Environment=CLAWDBOT_GATEWAY_PORT=18789
ExecStart=/usr/bin/node /usr/lib/node_modules/clawdbot/dist/index.js gateway
Restart=always
RestartSec=10
StandardOutput=journal
StandardError=journal

[Install]
WantedBy=multi-user.target
```

Enable and start:

```
sudo systemctl daemon-reload
sudo systemctl enable clawdbot
sudo systemctl start clawdbot
sudo systemctl status clawdbot
```

- Ignore `clawdbot doctor` complaints about the daemon; it only knows user services. Trust `systemctl status`.

Strategy B: Docker in LXC (Optional)

1. Proxmox LXC options

- Enable **nesting** and **keyctl** for the container.

2. Install Docker

```
curl -fsSL https://get.docker.com -o get-docker.sh
sh get-docker.sh
```

3. Docker Compose layout

```
mkdir -p ~/clawdbot-docker/config
mkdir -p ~/clawdbot-docker/workspace
```

`docker-compose.yml`:

```
services:
  clawdbot:
    image: ghcr.io/gsaraiva2109/clawdbot-build:latest
    container_name: clawdbot
    restart: unless-stopped
    network_mode: host
    environment:
      - HOME=/home/node
      - TERM=xterm-256color
      - CLAWDBOT_GATEWAY_BIND=0.0.0.0
      - CLAWDBOT_GATEWAY_PORT=18789
      - ANTHROPIC_API_KEY=${ANTHROPIC_API_KEY}
    volumes:
      - ./config:/home/node/.clawdbot
      - ./workspace:/home/node/clawd
```

- Uses host networking for mDNS discovery.

Networking & Access

- Clawdbot binds to `localhost` by default; for headless servers set `CLAWDBOT_GATEWAY_BIND=0.0.0.0`.
- For **secure onboarding**, don't expose 18789 publicly:
  - From your workstation:

```
ssh -L 18789:127.0.0.1:18789 user@your-proxmox-node
```

- Then open `http://127.0.0.1:18789` in a browser.

Onboarding & Config

- Web wizard lets you:
  - Name the agent, choose LLM provider, add API keys, link Telegram/WhatsApp.
- All config lives in:

```
~/.clawdbot/
```

- This directory is the "brain"; back it up.

Operating Clawdbot

- Logs (LXC service):

```
journalctl -u clawdbot -f
```

- Shows model calls, tool use, and gateway activity.
- Once wired to Telegram/WhatsApp, chats effectively act like a remote shell, so treat it as powerful and risky.

Security & Updates

Restrict who can talk to the bot

In `clawdbot.json`:

```
"telegram": {
  "enable": true,
  "token": "...",
```

```
  "allowFrom": [123456789]
}
```

- Without `allowFrom`, anyone who finds the bot can execute commands.

Updating

- LXC install:

```
npm install -g clawdbot@latest
systemctl restart clawdbot
```

- Docker:

```
docker compose pull
docker compose up -d
```

Big Picture

- Main challenge is not resources but Clawdbot's desktop assumptions vs. Proxmox's server/LXC environment.
- Once you run it as a system service (or Docker) with proper bind, SSH tunnel, and `allowFrom`, it becomes a stable, safe, always-on agent on your Proxmox box.