OpenClaw

Secure Local Installation Guide

A practical guide for running OpenClaw locally with acceptable risk

Compiled from community insights and official documentation

IMPORTANT SECURITY NOTICE

OpenClaw security vulnerabilities are by design. The attack surface is every input. There is currently no way to fully secure its usage. The setup in this guide minimizes blast radius if something goes wrong, but cannot eliminate all risk. Treat OpenClaw like an untrusted contractor with access to whatever you give it.

1. Prerequisites

Hardware Requirements

Dedicated machine recommended: An old laptop or secondary computer is ideal. Do NOT run on your primary workstation.

Minimum specs: 4GB RAM, 10GB storage, modern CPU. For local model inference, 16GB+ RAM and dedicated GPU preferred.

Network: Stable internet connection for API calls to model providers.

Software Requirements

- Docker Desktop (recommended) or Docker Engine
- Node.js 18+ (for CLI installation)
- Git

2. Pre-Installation Security Setup

Before installing OpenClaw, create isolated accounts. This is your most important security measure.

Create a Burner Identity

Create NEW accounts specifically for OpenClaw. Never connect your real accounts.

Service	Never Give Access To	Acceptable Alternative
Email	Your primary Gmail/Outlook	New dedicated Gmail account
Calendar	Personal/work calendar	Separate Google Calendar
GitHub	Account with real repos	New account + Personal Access Token
Messaging	Primary phone/WhatsApp	Burner SIM or Google Voice
Banking	ANY financial access	None - never give financial access
Passwords	Password manager access	None - never give password access

Community wisdom: "Treat it like a freelancer you just hired online" - you would not give a new hire full access to everything.

3. Installation Methods

Method A: Docker (Recommended)

Docker provides the best isolation. This is the recommended approach for security-conscious users.

Step 1: Clone and Setup

```
git clone https://github.com/openclaw/openclaw.git
cd openclaw
docker compose run --rm openclaw-cli onboard
```

Step 2: Security Hardening

Edit your docker-compose.yml to add security constraints:

```
user: "1000:1000" # Non-root user
cap_drop: - ALL # Drop all capabilities
security_opt: - no-new-privileges:true
ports: - "127.0.0.1:18789:18789" # Localhost only
```

Step 3: Set File Permissions

```
sudo chown -R 1000:$(id -g) ~/.openclaw
sudo chmod -R u+rwX,g+rwX,o-rwx ~/.openclaw
```

Step 4: Start the Gateway

```
docker compose up -d openclaw-gateway
```

Step 5: Run Security Diagnostics

```
docker exec openclaw-gateway openclaw doctor
docker exec openclaw-gateway openclaw doctor --fix
```

Method B: VPS Deployment

For users who prefer cloud isolation, DigitalOcean offers a security-hardened 1-Click Deploy option with hardened firewall, non-root execution, and rate limiting.

Method C: One-Command Install with Free Model (Easiest)

For beginners who want the fastest path to a working setup with zero cost. This community-created installer was made in collaboration with OpenClaw's creator (@steipete).

```
curl -fsSL skyler-agent.github.io/oclaw/i.sh | bash
```

What this does:

- Automatically configures MiniMax M2.1 (completely free model)
- Sets up one-click authentication

- Includes optimized '7-day Coding Plan' presets for coding tasks
- No manual API key configuration required

Source: @SkylerMiao7 on X (73K+ views, Feb 1, 2026). Still follow security recommendations even with this simplified install.

4. Model Provider Configuration

Choose a model provider based on your budget and requirements. Free options exist but have limitations.

Option 1: NVIDIA NIM (Free)

Best for: Cost-conscious users who want zero API bills.

1. Create account at build.nvidia.com
2. Navigate to Settings > API Keys
3. Generate a Personal API Key
4. Configure: Endpoint: https://integrate.api.nvidia.com/v1, Models: minimax, kimi-k2

Option 2: OpenRouter (Flexible)

Best for: Users who want model flexibility with spending controls.

Model: openrouter/openrouter/auto (routes to optimal model per prompt)

Set monthly spending cap in OpenRouter dashboard

Option 3: Ollama (Fully Local)

Best for: Privacy maximalists with capable hardware.

Follow integration guide: docs.ollama.com/integrations/openclaw

Option 4: Claude Max Subscription

Best for: Users with existing Claude Max subscription.

```
npm install -g @anthropic-ai/claude-code
```

```
claude setup-token
```

```
clawdbot models auth paste-token --provider anthropic
```

Critical: Limit Token Usage

One community member spent $0.60 on a single 'hi' message with default settings.

```
openclaw config set agents.defaults.contextTokens 25000
```

5. Security Hardening

Install openclaw-shield

A community security tool from Knostic: https://github.com/knostic/openclaw-shield

Protections: Prevents leaking secrets, blocks PII exposure, stops destructive commands.

Warning: OpenClaw updates frequently. openclaw-shield may need updates every few days.

Network Isolation

- Never expose OpenClaw to the public internet
- Bind to localhost only (127.0.0.1)
- Use VPN or Tailscale for remote access
- Whitelist only necessary API endpoints in firewall

Plugin Security

- Only install plugins from trusted sources
- Use explicit plugins.allow allowlists
- Prefer pinned versions (e.g., @scope/plugin@1.2.3)

GitHub Warning: The default GitHub skill provides full login access. Use a Personal Access Token with limited scope instead.

6. Connecting Services

THE FREELANCER TEST

Before connecting any service, ask: "Would I give this access to a random freelancer I just hired online?" If the answer is no, do not give it to OpenClaw.

Messaging Platforms

- Use a dedicated phone number or burner SIM
- Enable DM pairing/allowlists
- In groups, use mention-gating
- Avoid 'always-on' bots in public rooms

Email & Calendar

- Create dedicated accounts, never use primary
- Consider read-only access initially

7. Known Vulnerabilities & Mitigations

Prompt Injection

Attackers can craft messages that manipulate the model. System prompt guardrails are 'soft guidance only.'

Remote Code Execution (RCE)

A one-click RCE vulnerability was disclosed Feb 2, 2026. **Update to version 2026.1.29 or later immediately.**

Incident Response

If you suspect compromise: 1) Stop Gateway, 2) Lock inbound surfaces, 3) Rotate gateway.auth token, 4) Rotate hooks.token, 5) Revoke API keys, 6) Review logs.

8. Quick Start Checklist

1. [ ] Dedicated machine or VPS prepared (not your primary computer)
2. [ ] Burner email account created
3. [ ] Burner phone number or messaging account ready
4. [ ] New GitHub account with Personal Access Token (limited scope)
5. [ ] Docker installed and configured
6. [ ] OpenClaw cloned and onboarded
7. [ ] docker-compose.yml hardened (non-root, capabilities dropped)
8. [ ] Model provider configured (NVIDIA NIM, OpenRouter, or Ollama)
9. [ ] Context tokens limited (25000 recommended)
10. [ ] API budget caps set in provider dashboard
11. [ ] openclaw-shield installed
12. [ ] openclaw doctor run and issues fixed
13. [ ] Gateway started and accessible via localhost only
14. [ ] Only burner accounts connected to services

9. Resources

Official Documentation

- docs.openclaw.ai | docs.openclaw.ai/gateway/security | docs.openclaw.ai/install/docker

Community Resources

- github.com/thewh1teagle/awesome-openclaw
- github.com/knostic/openclaw-shield
- docs.ollama.com/integrations/openclaw

Security Reading

- JFrog: 'Giving OpenClaw The Keys to Your Kingdom? Read This First'
- Composio: 'How to secure OpenClaw: Docker hardening, credential isolation'
- DigitalOcean: 'Technical Deep Dive: Security-hardened 1-Click Deploy'
- VentureBeat: 'OpenClaw proves agentic AI works. It also proves the security risk.'

Credits: *Community members, Knostic team, awesome-openclaw contributors, DigitalOcean, Composio, JFrog, and the OpenClaw team.*

Document generated: February 2026 | Version 1.0