EN

Practical Guide

OpenClaw: Installation & First Agent

Workspace, skills and agent operational in 30 minutes

Mars 2026

Table of Contents

Thirty minutes. That's how long it takes, on a clean machine, to go from zero to an operational OpenClaw agent responding in your messaging app. This guide documents the exact path.

The previous post laid the conceptual foundations: what OpenClaw is, why self-hosted is becoming essential again, and how agent network logic transforms automation. This second installment gets concrete.

Prerequisites: What You Need Before Starting

Configuration validated by BOTUM teams:

• **OS**: Ubuntu 22.04 LTS or 24.04 LTS (recommended)

• **Node.js**: v18 or higher (node --version to check)

• **npm**: v9+ (bundled with Node.js)

• **Messaging access**: an account on your usual platform, with ability to create a bot

• **LLM API key**: Anthropic, OpenAI, or Google (depending on your model choice)

■ *A VPS with 2 vCPU / 2 GB RAM is more than sufficient for a production agent.*

1. Installation

```
npm install -g openclaw
```

Verify the command is accessible:

```
openclaw --version
```

Launch the initial configuration wizard:

```
openclaw init
```

The wizard asks which LLM to use, where to store the workspace, and some basic parameters. Allow two minutes.

2. The Workspace

The workspace is the heart of OpenClaw — the agent's persistent living space. By default in **~/.openclaw/workspace/**. It's a Git repository: every significant action is committed.

The fundamental files:

• **AGENTS.md** — The network map. Who does what, which agent is responsible for which domain.

• **SOUL.md** — The agent's identity. Its tone, values, and way of responding. Without SOUL.md, the agent responds generically.

• **MEMORY.md** — Curated long-term memory. User preferences, business rules, project context.

• **memory/YYYY-MM-DD.md** — Raw daily logs. Everything that happened in today's session.

■ *The power comes from this simple structure: the agent reads its files at each startup, giving persistent context without technical overhead.*

3. Connecting Messaging

```
openclaw gateway config
```

The wizard asks for messaging bot credentials. Once configured, start the gateway:

```
openclaw gateway start
```

The gateway listens to incoming messages and forwards them to the agent. In production, BOTUM teams configure it as a systemd service for automatic restart.

4. Activating First Skills

Weather skill — basic test

```
openclaw skills install weather
```

GitHub skill — for technical teams

```
openclaw skills install github
```

Each skill comes with a SKILL.md file the agent reads automatically. Progression principle: start simple, add skills as real needs emerge.

5. First Test: The Agent in Action

• **Message 1**: "What time is it?" — verifies messaging connection

• **Message 2**: "List the files in your workspace" — verifies filesystem access

• **Message 3**: "Create a test.md file with 'Hello, OpenClaw'" — verifies write permissions

■ *If all three responses are coherent, the installation is functional.*

Conclusion: The 30 Minutes Hold Up

Installation: 5 min • Configuration: 5 min • Messaging: 10 min • First skill: 5 min • Tests: 5 min = **30 minutes total**.

What takes time is what comes after: defining agent identities, structuring AGENTS.md for a network, configuring automations. That work is documented in the following posts.

Full article: blog.botum.ca/openclaw-install-workspace-first-agent

Website: www.botum.ca • contact@botum.ca