■ Guide Setup

Prépare ton environnement OpenClaw

Meetup IA · 5 mai 2026

Ce document te guide pas à pas pour installer et configurer OpenClaw sur ta machine. En le suivant, tu arriveras au meetup avec un agent opérationnel — prêt à travailler.

■ *Durée estimée : 30–60 min selon ton système. Fais ça la veille ou l'avant-veille.*

Table des matières

1. Prérequis système

1.1 Node.js 20+

OpenClaw tourne sur Node.js. Il te faut la version 20 minimum.

■ macOS

```
# Option 1 : Homebrew (recommandé)
brew install node

# Option 2 : nvm
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
nvm install 20
nvm use 20
```

■ Windows

```
# Option 1 : winget
winget install OpenJS.NodeJS.LTS

# Option 2 : télécharger
# → nodejs.org/en/download
# Choisir "LTS" (≥ 20.x)
# Lancer l'installateur .msi
```

■ Linux

```
# Ubuntu / Debian
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt-get install -y nodejs

# Arch Linux
sudo pacman -S nodejs npm
```

Vérification (toutes les plateformes) :

```
node --version # doit afficher v20.x ou plus
npm --version # doit afficher 9.x ou plus
```

■ *Si tu vois une version < 20, utilise nvm (Node Version Manager) pour changer de version sans désinstaller.*

1.2 Docker (optionnel — alternative à npm)

Si tu préfères éviter l'installation locale, OpenClaw peut tourner dans un container Docker.

■ macOS

```
# Télécharger Docker Desktop
# → docs.docker.com/desktop/install/mac-install

# Ou via Homebrew
brew install --cask docker
open /Applications/Docker.app
```

■ Windows

```
# Télécharger Docker Desktop
# → docs.docker.com/desktop/install/windows-install

# Prérequis Windows
# → WSL2 activé
# → Virtualisation BIOS ON
```

■ Linux

```
# Ubuntu
curl -fsSL https://get.docker.com | sh
sudo usermod -aG docker $USER
newgrp docker

# Vérifier
docker --version
```

2. Installation OpenClaw

2.1 Via npm (méthode standard)

```
# Installation globale
npm install -g openclaw

# Vérifier l'installation
openclaw --version

# Lancer le wizard de configuration
openclaw onboard
```

Le wizard onboard te posera 4 questions :

1	Quel LLM veux-tu utiliser ? (Claude / GPT / DeepSeek / autre)
2	Colle ta clé API
3	Canal de messagerie ? (Telegram recommandé pour débuter)
4	Token du bot Telegram

2.2 Via Docker (sans Node.js local)

```
# Créer un dossier pour ton agent
mkdir mon-agent && cd mon-agent

# Lancer OpenClaw dans Docker
docker run -it --rm \
-v $(pwd):/workspace \
-e ANTHROPIC_API_KEY=ta_cle_api \
-p 18789:18789 \
ghcr.io/openclaw/openclaw:latest
```

■ *Docker = zéro installation Node.js. Idéal si tu as des conflits de version ou que tu veux tester rapidement.*

2.3 Structure des fichiers créés

```
mon-agent/
■■■ SOUL.md ← personnalité de ton agent
■■■ MEMORY.md ← mémoire long terme
■■■ AGENTS.md ← instructions de démarrage
■■■ CODEX.md ← état technique
■■■ openclaw.json ← config principale (LLM, canaux)
■■■ memory/ ← logs quotidiens
■■■ scripts/ ← tes scripts
■■■ data/ ← fichiers de ton agent
```

3. Créer tes comptes LLM

Choisis au moins UN fournisseur. Tu peux en configurer plusieurs et switcher à la volée.

3.1 Anthropic — Claude (recommandé)

1. Aller sur **console.anthropic.com** → "Sign up"
2. Créer un compte (Google ou email)
3. Aller dans **Settings** → **API Keys** → "Create key"
4. Copier la clé (commence par **sk-ant-api03-...**)
5. Ajouter 5$ de crédit minimum dans **Billing** → **Plans**

```
# Tester ta clé (curl)
curl https://api.anthropic.com/v1/messages \
-H "x-api-key: sk-ant-api03-VOTRE_CLE" \
-H "anthropic-version: 2023-06-01" \
-H "content-type: application/json" \
-d '{"model":"claude-3-5-haiku-20241022","max_tokens":50,"messages":[{"role":"user","content":"Bonjour"}]}'
```

■ *5$ = environ 5 millions de tokens. Largement suffisant pour le meetup.*

3.2 DeepSeek (modèle puissant, très économique)

1. Aller sur **platform.deepseek.com** → "Sign Up"
2. Créer un compte (Google ou email)
3. Aller dans **API Keys** → "Create new API key"
4. Copier la clé (commence par **sk-...**)
5. Recharger 2-5$ de crédit (recharges min 2$)

■ *DeepSeek V3 = performances proches de Claude à 10× moins cher. Idéal pour les tâches répétitives.*

3.3 OpenRouter (accès 100+ modèles via une seule clé)

1. Aller sur **openrouter.ai** → "Sign In"
2. Créer un compte (GitHub recommandé)
3. Aller dans **Keys** → "Create Key"
4. Copier la clé (commence par **sk-or-v1-...**)
5. Créditer ton compte (min 5$)

Modèles disponibles via OpenRouter :

Modèle	Fournisseur	Usage recommandé
anthropic/claude-sonnet-4-5	Anthropic	Tâches complexes, raisonnement
deepseek/deepseek-chat	DeepSeek	Tâches répétitives, économique
meta-llama/llama-3.1-70b	Meta (via OR)	Open source, polyvalent
google/gemini-2.5-flash	Google	Rapide, multimodal, gratuit*
mistralai/mistral-7b	Mistral	Léger, local-compatible

3.4 Ollama — Modèles locaux (zéro cloud, zéro coût)

Ollama te permet de faire tourner Llama, Mistral, Gemma et d'autres en local. Aucune clé API, aucun coût, données qui ne quittent jamais ta machine.

■ macOS

```
# Télécharger Ollama
# → ollama.ai → Download → macOS
# (application .dmg)

# Après installation
ollama pull llama3.2
ollama run llama3.2
```

■ Windows

```
# Télécharger
# → ollama.ai → Download → Windows
# Lancer OllamaSetup.exe

# Dans PowerShell
ollama pull llama3.2
ollama run llama3.2
```

■ Linux

```
# Installation one-liner
curl -fsSL https://ollama.ai/install.sh | sh

# Télécharger un modèle
ollama pull llama3.2

# Lancer
ollama run llama3.2
```

Modèles locaux recommandés (tous gratuits, tous via Ollama) :

Modèle	Taille	RAM min	Points forts
llama3.2	3B	4 GB	Léger, rapide, parfait pour débuter
llama3.1:8b	8B	8 GB	Équilibre performance/vitesse

qwen2.5:7b	7B	8 GB	Excellent en code + multilangue (chinois inclus)
qwen2.5:14b	14B	16 GB	Très performant, comparable à GPT-3.5
qwen2.5-coder:7b	7B	8 GB	Spécialisé coding — très précis
mistral:7b	7B	8 GB	Européen, open source, généraliste
phi4:14b	14B	16 GB	Microsoft — raisonnement avancé
gemma3:9b	9B	10 GB	Google — léger et multimodal
deepseek-r1:7b	7B	8 GB	Raisonnement, maths, très performant

```
# Installer n'importe quel modèle Ollama
ollama pull qwen2.5:7b
ollama pull mistral:7b
ollama pull deepseek-r1:7b

# Lister tes modèles installés
ollama list

# Configurer dans openclaw.json
"model": "ollama/qwen2.5:7b"
```

■■ Souveraineté numérique — Pourquoi ça change tout

Choisir entre cloud et local, c'est choisir qui contrôle tes données. Ce tableau t'aide à comprendre l'impact réel.

	■■ Cloud (Anthropic, OpenAI...)	■ Local (Ollama sur ta machine)
Tes données	Transitent sur des serveurs US/EU	Ne quittent JAMAIS ta machine
RGPD / Loi 25	À vérifier selon le contrat	■ Conforme par défaut
Confidentialité	Données potentiellement utilisées pour training	■ Zéro collecte
Coût	Facturation à l'usage (tokens)	■ Gratuit (électricité seulement)
Performance	■■■■■ Meilleure qualité	■■■ Bonne, en progression rapide
Disponibilité	Dépend d'internet + uptime provider	■ 100% offline possible
Usage pro sensible	■■ Vérifier les politiques	■ Recommandé (RH, légal, finance)

■ *Recommandation : utilise Claude ou DeepSeek pour tes tâches quotidiennes, et Ollama (Qwen ou Mistral) pour tout ce qui est confidentiel — RH, données clients, finance.*

4. Configurer ton LLM dans OpenClaw

Le fichier **openclaw.json** est la configuration principale. Il se trouve dans ton dossier workspace.

Exemple — Claude (Anthropic) :

```
{
"model": "anthropic/claude-sonnet-4-5",
"apiKeys": {
"anthropic": "sk-ant-api03-TON_API_KEY"
}
}
```

Exemple — DeepSeek via OpenRouter :

```
{
"model": "deepseek/deepseek-chat",
"apiKeys": {
"openrouter": "sk-or-v1-TON_API_KEY_OPENROUTER"
}
}
```

Exemple — Ollama (local, aucune clé) :

```
{
"model": "ollama/llama3.2",
"ollama": {
"host": "http://localhost:11434"
}
}
```

■ *Tu peux changer de modèle à chaud sans redémarrer : édite openclaw.json et envoie "reload config" à ton agent.*

5. Connecter Telegram

Telegram est le canal le plus simple pour démarrer. Tu crées un bot en 2 minutes avec @BotFather.

1	Ouvrir Telegram → chercher **@BotFather** → cliquer "Start"
2	Envoyer **/newbot**
3	Choisir un nom affiché (ex: "Mon Agent IA")
4	Choisir un username qui finit par "bot" (ex: **monagent_bot**)
5	Copier le token (ressemble à : **1234567890:AAFxx...**)
6	Dans openclaw.json → ajouter le token dans la section "telegram"
7	Lancer **openclaw start** → envoyer "Bonjour" à ton bot

Config telegram dans openclaw.json :

```
"channels": {
"telegram": {
"token": "1234567890:AAFxx_TON_TOKEN_ICI",
"allowedSenders": ["TON_TELEGRAM_USER_ID"]
}
}
```

■ *Pour trouver ton Telegram User ID : envoyer un message à @userinfobot. Il te répond avec ton ID numérique.*

6. Nano Banana Pro — Génération d'images

Nano Banana Pro est un skill OpenClaw qui génère des images avec Gemini Image. Il nécessite une clé Google AI Studio (gratuit).

Créer ta clé Google AI Studio :

1	Aller sur **aistudio.google.com**
2	Connecter ton compte Google
3	Cliquer **"Get API key"** → "Create API key"
4	Copier la clé (commence par **AIzaSy...**)

Installer et activer le skill :

```
# Installer nano-banana-pro
clawhub install nano-banana-pro

# Ajouter ta clé dans openclaw.json
"apiKeys": {
"gemini": "AIzaSy_TON_API_KEY"
}

# Tester
# → Envoie à ton agent : "génère une image d'un robot qui code"
```

7. HERMÈS — Agent Email

HERMÈS gère ta boîte Gmail : tri, digest, réponses automatiques. Il utilise himalaya (CLI email) avec un "app password" Gmail.

Créer un app password Gmail :

1	Aller sur **myaccount.google.com/security**
2	Activer la **validation en 2 étapes** (obligatoire)
3	Chercher **"Mots de passe des applications"**
4	Créer un nouveau → Nom : "openclaw" → Générer
5	Copier le code à 16 caractères (ex: **abcd efgh ijkl mnop**)

Config himalaya (compte gmail) :

```
# ~/.config/himalaya/config.toml
[accounts.gmail]
email = "toi@gmail.com"
display-name = "Ton Nom"

[accounts.gmail.incoming]
type = "imap"
host = "imap.gmail.com"
port = 993
login = "toi@gmail.com"
passwd.cmd = "echo 'abcd efgh ijkl mnop'"

[accounts.gmail.outgoing]
type = "smtp"
host = "smtp.gmail.com"
port = 587
login = "toi@gmail.com"
passwd.cmd = "echo 'abcd efgh ijkl mnop'"
```

■ *Pour la sécurité, stocke le mot de passe dans 1Password ou Vault et utilise la commande op read au lieu de echo.*

8. Personnaliser ton agent — SOUL.md & MEMORY.md

SOUL.md — La personnalité

SOUL.md définit QUI est ton agent : sa façon de parler, ses valeurs, ses limites. C'est la première chose qu'il lit à chaque démarrage.

```
# SOUL.md — exemple minimal

## Qui je suis
Je m'appelle [Nom]. Je suis l'assistant IA de [ton prénom].

## Comment je communique
- Direct et concis — pas de fla-fla
- En français sauf si on me parle anglais
- J'ai le droit d'avoir une opinion

## Ce que je fais bien
- Email, recherche web, gestion de fichiers
- Scripting Python et bash
- Organisation et rappels

## Mes limites
- Toujours demander avant d'envoyer un email à un tiers
- Ne jamais supprimer de fichiers sans confirmation
```

MEMORY.md — La mémoire long terme

MEMORY.md stocke ce que ton agent doit toujours garder en tête : tes préférences, tes contacts importants, tes règles.

```
# MEMORY.md — exemple minimal

## Règles absolues
- TOUJOURS demander avant d'envoyer un email
- JAMAIS supprimer un fichier sans GO

## Mes préférences
- Langue : français (sauf emails pros en anglais)
- Ton : direct, pas de "Bien sûr !" ni "Absolument !"

## Contacts importants
- Mon manager : [nom]
- Mon équipe : [noms]

## Navigation rapide
- Logs quotidiens → memory/YYYY-MM-DD.md
```

9. Checklist finale — Avant le meetup

Coche chaque item. Si tout est vert, tu arrives prêt(e) pour créer ton premier agent !

☐	**Node.js 20+ installé**	*node --version → v20.x ou plus*
☐	**OpenClaw installé**	*openclaw --version → 2026.x.x*
☐	**Au moins 1 compte LLM créé**	*Anthropic, DeepSeek ou OpenRouter*
☐	**Clé API configurée**	*openclaw.json → apiKeys rempli*
☐	**Bot Telegram créé**	*@BotFather → token copié*
☐	**Telegram connecté**	*openclaw start → "Bonjour" → réponse reçue*
☐	**SOUL.md personnalisé**	*Au moins ton nom + 3 règles*
☐	**MEMORY.md initialisé**	*Quelques préférences notées*
☐	**Test complet réussi**	*Envoyer "résume les news tech du jour" → réponse*

Ressources & aide

Repo GitHub

github.com/openclaw/openclaw

Discord

discord.gg/clawd

Des questions ? Rejoins le Discord avant le meetup — la communauté répond rapidement. À mardi !

10. Premier lancement — tout assembler

Cette section résume la config complète. Suis les étapes dans l'ordre : à la fin, ton agent tourne et répond sur Telegram.

Le fichier openclaw.json complet (exemple Claude + Telegram) :

```
{
"model": "anthropic/claude-sonnet-4-5",
"apiKeys": {
"anthropic": "sk-ant-api03-REMPLACE_PAR_TA_CLE"
},
"channels": {
"telegram": {
"token": "1234567890:AAFxx_REMPLACE_PAR_TON_TOKEN",
"allowedSenders": ["TON_TELEGRAM_USER_ID"]
}
},
"workspace": "./",
"sessionKey": "main"
}
```

■ *Pour OpenRouter : remplace "anthropic" par "openrouter" et la clé par ta clé OR. Pour Ollama : retire apiKeys et ajoute "ollama": {"host": "http://localhost:11434"}.*

Séquence de démarrage — 7 étapes

1 **Ouvrir un terminal dans ton dossier workspace**

```
cd ~/mon-agent (ou le dossier créé par openclaw onboard)
```

2 **Vérifier que openclaw.json est complet**

```
cat openclaw.json → vérifie model, apiKeys, channels
```

3 **Si Ollama : s'assurer qu'il tourne**

```
ollama serve (dans un terminal séparé)
```

4 **Lancer OpenClaw**

```
openclaw start
```

5 **Ouvrir Telegram → trouver ton bot → cliquer "Start"**

```
Envoyer : "Bonjour"
```

6 **Réponse attendue dans < 5 secondes**

```
"Bonjour ! Comment puis-je t'aider ?" ← ton agent tourne !
```

7 **Tester une vraie commande**

```
"Résume les actualités tech du jour" ou "Quelle heure est-il ?"
```

Problèmes courants

■ **"API key invalid"**	Vérifie que la clé est bien copiée sans espace. Essaie de la recréer.
■ **"Bot not found" Telegram**	Le token est mal copié. Retourne sur @BotFather → /mybots → API token.
■ **"Unauthorized sender"**	Ton User ID n'est pas dans allowedSenders. Vérifie via @userinfobot.
■ **Ollama ne répond pas**	Lance ollama serve dans un terminal séparé avant openclaw start.
■ **Aucune réponse > 30 sec**	Regarde les logs : openclaw logs → cherche "error" ou "timeout".

■ *Ton agent tourne ? Parfait. Apporte ton laptop mardi — on pousse ça encore plus loin ensemble !*