OpenClaw

Guide d'installation et de configuration

VPS Ubuntu/Debian · Architecture · Bonnes pratiques

Mars 2026

1. Architecture d'OpenClaw

1.1 Les deux couches de configuration

OpenClaw repose sur deux couches distinctes que vous ne modifiez pas de la même façon.

Couche	Rôle
~/clawd/	Votre espace de travail personnel : configuration, mémoire, compétences, secrets. C'est ici que vous personnalisez.
/opt/.../node_modules/clawdbot/	Installation globale : comportements par défaut, compétences intégrées. Se met à jour via npm update -g openclaw.

⚠ Ne touchez qu'à ~/clawd/. L'installation globale ne doit être modifiée que via npm update.

1.2 Structure de l'espace de travail

Voici l'organisation complète du dossier ~/clawd/ avec le rôle de chaque fichier :

```
~/clawd/
├── AGENTS.md          → instructions de fonctionnement (priorité maximale)
├── SOUL.md            → personnalité et ton de l'agent
├── IDENTITY.md        → identité de l'agent (nom, ambiance)
├── USER.md            → votre profil (timezone, préférences)
├── HEARTBEAT.md       → vérifications planifiées et tâches cron
├── TOOLS.md           → documentation des outils externes
├── BOOTSTRAP.md       → rituel de première session (supprimer après)
├── skills/            → workflows réutilisables
│   ├── meeting-prep/
│   └── social-post-writer/
├── memory/            → contexte persistant
│   ├── 2026-03-22.md
│   └── key-context.md
├── .secrets/          → clés API (jamais committé, jamais versionné)
│   ├── anthropic-api-key.txt
│   └── task-api-token.txt
└── output/            → fichiers générés et exports
```

1.3 Rôle détaillé de chaque fichier

AGENTS.md — Le fichier le plus important

Chargé en prompt système à chaque démarrage de session. Tout ce qui est écrit ici s'applique en permanence pendant toute la conversation. C'est votre manuel d'instructions opérationnel.

Ce qu'il doit contenir : directive principale, outils clés et comment les utiliser, règles strictes à ne jamais enfreindre, déclencheurs de workflow.

Ce qu'il ne doit PAS contenir : documentation complète (faites des références), longs paragraphes théoriques, informations susceptibles de changer souvent.

> **Limite critique**
> Garder AGENTS.md sous 300 lignes. Au-delà, le fichier consomme trop de contexte et l'obéissance aux instructions diminue significativement.

```
# AGENTS.md - Exemple minimal efficace

## Directive opérationnelle principale
Mon assistant fonctionne comme un aide proactif.

## Outils clés
- Email : vérifier la boîte quotidiennement, signaler les urgences
- Calendrier : vérifier les conflits avant de planifier

## Règles strictes
- NE JAMAIS redémarrer les services sans autorisation explicite
- Toujours vérifier la date avec `date` avant de planifier
```

SOUL.md — Personnalité et limites

Définit qui est votre agent et comment il communique. Se construit par itération : ajoutez une ligne 'ne fais jamais X' chaque fois que quelque chose vous agace. Les règles négatives ('ne fais jamais X') fonctionnent mieux que les règles positives ('essaie d'être Y').

```
# SOUL.md - Exemple

Tu es [nom]. Tu assistes [ton nom].
Sois direct. Pas de remplissage. Égalise mon ton.
Si je pose une question, réponds d'abord. Développe seulement si besoin.
Ne dis jamais 'absolument', 'super question' ou 'je serais ravi de le faire.'
Si tu ne sais pas quelque chose, dis-le. Ne devine pas.
Si une tâche va coûter des tokens significatifs, dis-le avant de la faire.
```

IDENTITY.md et USER.md — Qui est qui

IDENTITY.md définit votre agent (nom, ambiance). Change rarement une fois fixé.

USER.md vous définit (timezone, téléphone, calendrier, préférences). Se met à jour à mesure que vos préférences évoluent.

```
# IDENTITY.md
```

```
- Nom : [Nom de votre agent]
- Ambiance : Intelligent, utile, fiable

# USER.md
- Nom : [Votre nom]
- Fuseau horaire : Europe/Paris
- Calendrier : Google Calendar
```

HEARTBEAT.md — Vérifications planifiées

Contrôle ce que fait l'agent lors des sondages automatiques. Permet la proactivité : vérifier le calendrier, scanner la boîte mail, exécuter des workflows selon un planning.

```
# HEARTBEAT.md

Si le message contient 'EXECUTION CRON' :
  - Lire les instructions complètes dans le message
  - Exécuter toutes les étapes requises
  - Ne jamais retourner HEARTBEAT_OK

Sinon :
  - Si rien ne nécessite attention → retourner HEARTBEAT_OK
  - Garder cela minimal
```

TOOLS.md — Documentation des outils externes

Ne définit pas quels outils existent (OpenClaw gère cela en interne), mais documente comment vous les utilisez spécifiquement. Conventions, tokens, comportements particuliers.

skills/ — Workflows réutilisables

Chaque compétence est un dossier autonome avec un fichier SKILL.md. L'agent l'invoque quand la tâche correspond à la description. Les fichiers de support ne se chargent que lors de l'activation de la compétence.

```
skills/
├── meeting-prep/
│   └── SKILL.md
└── social-post-writer/
    ├── SKILL.md
    └── references/templates.md

# Exemple de frontmatter SKILL.md
---
name: meeting-prep
description: Recherche et briefing préalable. Utiliser quand l'utilisateur
             demande de préparer une réunion à venir.
---
```

memory/ — Contexte persistant

Mémoire long terme de l'agent. Sans fichiers de mémoire, chaque conversation repart de zéro.

- Format recommandé : memory/YYYY-MM-DD.md pour les journaux quotidiens
- L'agent lit automatiquement aujourd'hui et hier au démarrage de session
- Capturer faits, préférences et décisions durables
- Ne jamais stocker de secrets ici

Architecture VPS recommandée

```
VPS hôte
├── openclaw (systemd, natif)   ← agent principal
│   ├── accès apt, systemctl, fichiers système
│   └── pilote Docker depuis l'extérieur
│
├── app1 (conteneur Docker)     ← apps déployées par l'agent
├── app2 (conteneur Docker)
└── nginx (reverse proxy)
```

OpenClaw tourne nativement sur le VPS pour avoir un accès système complet. Les applications qu'il déploie tournent dans Docker.

2. Installation sur VPS

Prérequis
VPS Ubuntu 22.04/24.04 ou Debian 12 · Minimum 2 Go RAM / 2 vCPU / 20 Go disque · Accès root SSH · Clé API Anthropic · Bot Telegram ou Discord

2.1 Créer un user dédié

⚠ Ne jamais lancer OpenClaw en root en production.

```
useradd -m -s /bin/bash openclaw
passwd openclaw

# Sudo ciblé (recommandé, plus sécurisé que sudo complet)
visudo -f /etc/sudoers.d/openclaw
```

Contenu du fichier sudoers (option B recommandée) :

```
openclaw ALL=(ALL) NOPASSWD: /usr/bin/apt, /usr/bin/apt-get,
  /bin/systemctl, /usr/bin/docker, /usr/bin/npm, /usr/bin/pip3,
  /bin/mkdir, /bin/cp, /bin/mv, /bin/rm, /usr/sbin/nginx
```

```
# Ajouter aux groupes nécessaires
usermod -aG docker openclaw
usermod -aG systemd-journal openclaw
```

2.2 Installer Node.js via nvm

⚠ Ne pas utiliser le script NodeSource — il nécessite des droits root non disponibles pour l'user dédié.

```
su - openclaw

# Installer nvm
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash

# Recharger le shell
source ~/.bashrc

# Installer Node.js 22
nvm install 22
nvm use 22
nvm alias default 22
```

```
# Vérifier
node --version   # doit afficher v22.x.x
npm --version
```

2.3 Installer et configurer OpenClaw

```
npm install -g pnpm
npm install -g openclaw

# Lancer l'onboarding
openclaw onboard
```

L'onboarding configure dans l'ordre :

1. Anthropic Token — clé API depuis console.anthropic.com
2. Modèle recommandé — voir section Coûts
3. Search provider — Tavily (gratuit, 1000 req/mois) ou DuckDuckGo (sans clé)
4. Canal de communication — token Telegram ou Discord

2.4 Configurer le service systemd

> **Pourquoi systemd système ?**
> Les systemd user services sont souvent indisponibles sur VPS. Utiliser un service système à la place.

```
# Trouver le chemin exact de Node
which openclaw
# Exemple : /home/openclaw/.nvm/versions/node/v22.22.2/bin/openclaw

sudo nano /etc/systemd/system/openclaw-gateway.service
```

```
[Unit]
Description=OpenClaw Gateway
After=network.target

[Service]
Type=simple
User=openclaw
WorkingDirectory=/home/openclaw
ExecStart=/home/openclaw/.nvm/versions/node/v22.22.2/bin/openclaw gateway
Restart=always
RestartSec=5
Environment=HOME=/home/openclaw
Environment=PATH=/home/openclaw/.nvm/versions/node/v22.22.2/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin

[Install]
```

```
WantedBy=multi-user.target
```

⚠ La ligne Environment=PATH est obligatoire. Sans elle, systemd utilise le Node système (v20) au lieu du v22 installé via nvm, ce qui fait crasher le service.

```
sudo systemctl daemon-reload
sudo systemctl enable openclaw-gateway
sudo systemctl start openclaw-gateway
sudo systemctl status openclaw-gateway
```

2.5 Commandes quotidiennes

Action	Commande
Voir les logs en direct	sudo journalctl -u openclaw-gateway -f
Redémarrer le service	sudo systemctl restart openclaw-gateway
Stopper le service	sudo systemctl stop openclaw-gateway
Statut de l'agent	openclaw status
Diagnostic complet	openclaw doctor
Mettre à jour	npm update -g openclaw

3. Sécurité — À faire avant tout

3.1 Verrouiller la gateway

C'est le point le plus critique. Vérifier immédiatement après installation :

```
openclaw config get | grep host
```

⚠ Si le résultat est 0.0.0.0 ou vide, votre agent est exposé publiquement. N'importe qui sur internet qui trouve votre IP peut lui envoyer des messages — et votre agent aura accès à votre email et calendrier.

Corriger immédiatement :

```
{ "gateway": { "host": "127.0.0.1" } }
```

Puis accéder via tunnel SSH uniquement :

```
ssh -L 18789:localhost:18789 user@votre-vps
```

Temps nécessaire
2 minutes. À faire maintenant — pas après avoir configuré Telegram. Maintenant.

3.2 Autres règles de sécurité

- Activer UFW : ufw allow 22,80,443/tcp && ufw enable
- Protéger le token Telegram/Discord — c'est la clé d'accès à votre VPS
- Surveiller les logs régulièrement : sudo journalctl -u openclaw-gateway
- Le dossier .secrets/ doit être gitignored et jamais commité

3.3 Dépannage fréquent

Symptôme	Cause probable	Solution
Gateway not reachable	Service non démarré	openclaw doctor → openclaw gateway start
Node.js version error (v20)	PATH manquant dans systemd	Ajouter Environment=PATH=... avec chemin nvm complet
sudo: command not found	Chemin incorrect dans sudoers	Vérifier /etc/sudoers.d/openclaw, adapter les chemins
Permission denied	Groupe docker non attribué	usermod -aG docker openclaw puis se

sur Docker		reconnecter

4. Modèle et coûts

⚠ Changer le modèle par défaut AVANT de faire quoi que ce soit. Si vous n'avez pas touché à ce paramètre, vous utilisez probablement Opus — le modèle le plus cher.

4.1 Changer le modèle

Passer sur Sonnet. Vous ne remarquerez pas la différence pour les tâches normales. Vous remarquerez la différence sur la facture.

```
{ "ai": { "model": "claude-sonnet-4-5-20250929" } }
```

4.2 Coûts de référence

Configuration	Coût mensuel estimé
Sonnet + 1 agent + sans compétences (usage modéré)	3–8 $/mois
Opus sans optimisation	jusqu'à 47 $/semaine
Sonnet + même usage qu'Opus	~6 $/semaine → économie ×7

Si vous dépensez plus de 10 $ la première semaine avec un seul agent sur Sonnet, quelque chose ne va pas — et c'est réparable.

4.3 La commande /new — la plus sous-estimée

Chaque message que vous envoyez dans une session est inclus dans chaque futur appel API. Après une semaine de conversation, vous envoyez des milliers de tokens d'anciennes conversations avec chaque nouveau message.

/new vide le buffer de conversation sans effacer la mémoire. L'agent conserve tous ses fichiers, SOUL.md, tout.

Quand utiliser /new

Avant toute grosse tâche (recherche, rédaction, analyse) · Quand l'agent commence à agir bizarrement · Au moins une fois par jour comme habitude

5. Bonnes pratiques — Première semaine

5.1 Vue d'ensemble

	À faire	À éviter
Modèle	Passer sur Sonnet immédiatement	Laisser Opus par défaut
Démarrage	Commencer par BOOTSTRAP.md et SOUL.md	Commencer par une vraie tâche
Sessions	Utiliser /new régulièrement	Laisser une session s'accumuler sur des jours
Compétences	Ajouter une à la fois après stabilisation	Installer des skills depuis clawhub dès le départ
Agents	Un seul pendant 2 semaines minimum	Créer plusieurs agents pour 'tout couvrir'
Coûts	Vérifier quotidiennement les 2 premières semaines	Découvrir la facture en fin de mois

5.2 Planning jour par jour

```
Jours 1-2 : Configurer SOUL.md, avoir des conversations normales,
            poser des questions idiotes, se mettre à l'aise.

Jours 3-4 : Commencer à utiliser pour des tâches réelles.
            Calendrier, rappels, recherches web, résumés d'articles.
            Les choses ennuyeuses.

Jours 5-7 : Affiner SOUL.md en fonction de ce qui vous a agacé.
            Vérifier les coûts. Prendre conscience de votre
            consommation quotidienne.
```

5.3 Attention aux skills clawhub

⚠ Certaines compétences s'exécutent silencieusement et brûlent des tokens en arrière-plan. Certaines gonflent la fenêtre de contexte. Des centaines ont été signalées comme malveillantes (infostealers, backdoors).

- Ne rien installer la première semaine
- Après stabilisation : ajouter une seule compétence à la fois
- Tester chaque nouvelle compétence quelques jours avant d'en ajouter une autre

- Jamais plus d'une compétence à la fois

5.4 Pourquoi un seul agent suffit

Tous les nouveaux utilisateurs pensent qu'ils ont besoin de plusieurs agents. Un pour le personnel, un pour le travail, un pour le code. Ce n'est pas le cas — pas encore.

- Chaque agent est un consommateur de tokens indépendant
- Chaque agent a besoin de son propre lien de canal
- Chaque agent complique le débogage
- Créer un deuxième agent pour 'réparer' le premier donne deux agents cassés

> **La règle**
> Faire fonctionner un agent parfaitement pendant 2 semaines. Ensuite décider si vous avez vraiment besoin d'un deuxième. La plupart des gens n'en ont pas.

5.5 Après la première semaine

Si votre agent semble utile, que vos coûts sont inférieurs à 10 $ et que rien ne se casse aléatoirement, vous êtes prêt à expérimenter.

5. Ajouter la recherche web si ce n'est pas déjà fait
6. Puis une compétence de briefing quotidien
7. Puis peut-être une intégration calendrier pour des rappels proactifs

Construire lentement. Gagner chaque nouvelle capacité en s'assurant que la précédente est stable d'abord.

Les gens qui survivent au premier mois sont ceux qui ont commencé ennuyeux.