

HIERO DIGITAL INTERNATIONAL

OpenClaw

Déploiement gateway · Configuration · Sécurisation

Document technique interne — équipes HIERO

Version OpenClaw de référence : 2026.6.x

Diffusion restreinte — usage interne

0. Objet & prérequis

Ce playbook guide un technicien HIERO pour déployer et sécuriser OpenClaw Gateway sur un terminal Android. Il couvre l'installation de l'environnement d'exécution (Termux), la configuration minimale du Gateway, le premier lancement, et la sécurisation initiale.

i OpenClaw sur Android s'appuie sur Termux (émulateur de terminal Linux). Aucune émulation ARM ou root n'est requise pour une installation standard.

Prérequis matériels et logiciels

Élément	Minimum requis	Recommandé
Android	10 (API 29)	12+ (API 31)
RAM disponible	2 Go libres	4 Go
Stockage	1 Go libre	4 Go
Connexion	Wi-Fi stable	Wi-Fi + données mobiles
Termux	0.118+ (F-Droid)	Dernière version F-Droid
Node.js	v20 LTS	v22 LTS

⚠ Installer Termux uniquement depuis F-Droid (<https://f-droid.org>). La version Google Play est obsolète et non maintenue.

1. Installer Termux

1.1 Télécharger et installer

1. Ouvrir le navigateur Android et aller sur <https://f-droid.org>
2. Télécharger l'APK F-Droid et l'installer (autoriser les sources inconnues si demandé)
3. Depuis l'app F-Droid, rechercher « Termux » et installer la dernière version stable
4. Lancer Termux — un terminal Linux s'ouvre

1.2 Mise à jour initiale

Exécuter impérativement ces commandes à la première ouverture :

```
pkg update && pkg upgrade -y
```

i Cette étape peut prendre 3 à 10 minutes selon la connexion. Ne pas interrompre.

1.3 Activer le stockage partagé (optionnel)

Nécessaire si vous souhaitez accéder aux fichiers Android depuis Termux :

```
termux-setup-storage
```

Accepter la permission de stockage dans la popup Android. Le dossier ~/storage/ sera créé.

2. Installer Node.js et les dépendances

2.1 Node.js

```
pkg install nodejs-lts -y
```

Vérifier l'installation :

```
node --version    # doit afficher v20.x.x ou v22.x.x
npm --version     # doit afficher 10.x.x ou supérieur
```

2.2 Dépendances système

```
pkg install git curl wget openssl-tool -y
```

i git est requis pour versionner votre workspace. openssl-tool est utilisé pour la génération de jetons Gateway.

2.3 Dépendances optionnelles

Selon l'usage prévu :

- Python 3 (pour les scripts utilitaires) : `pkg install python -y`
- jq (manipulation JSON en CLI) : `pkg install jq -y`
- SSH (accès distant sécurisé) : `pkg install openssh -y`

3. Installer OpenClaw

3.1 Installation via npm

```
npm install -g openclaw
```

Vérifier l'installation :

```
openclaw --version
```

⚠ Si npm global échoue avec une erreur de permissions, configurer le prefix npm dans le home Termux : `mkdir -p ~/.npm-global` `npm config set prefix '~/.npm-global'` `export PATH=~/.npm-global/bin:$PATH` Ajouter la ligne `export PATH` dans `~/.bashrc` pour la rendre persistante.

3.2 Vérifier l'état initial

```
openclaw doctor
```

Cette commande détecte les problèmes de configuration et de permissions. Corriger toute erreur signalée avant de continuer.

4. Configuration initiale du Gateway

4.1 Générer le jeton Gateway

```
openclaw doctor --generate-gateway-token
```

Copier le jeton affiché — il ne sera plus visible. Le traiter comme un secret d'administration.

4.2 Créer openclaw.json

Créer et éditer le fichier de configuration :

```
mkdir -p ~/.openclaw  
nano ~/.openclaw/openclaw.json
```

Coller la configuration minimale suivante (remplacer <JETON> par la valeur générée) :

```
{  
  gateway: {  
    mode: "local",  
    bind: "loopback",  
    port: 18789,  
    auth: { mode: "token", token: "<JETON>" },  
  },  
  agents: {  
    defaults: {  
      workspace: "~/.openclaw/workspace",  
      model: { primary: "mistral/mistral-large-latest" },  
      heartbeat: { every: "0m" },  
    },  
    list: [ { id: "main", default: true } ],  
  },  
  session: { dmScope: "per-channel-peer" },  
}
```

⚠ bind: loopback est impératif. Sur Android, le terminal est accessible localement. N'exposez jamais 0.0.0.0 sans tunnel authentifié.

4.3 Resserrer les permissions

```
chmod 700 ~/.openclaw
chmod 600 ~/.openclaw/openclaw.json
```

5. Premier démarrage

5.1 Lancer le Gateway

```
openclaw gateway start
```

Le Gateway démarre et écoute sur 127.0.0.1:18789. La sortie doit indiquer :

```
✓ Gateway running on loopback:18789
```

5.2 Vérifier la santé

Dans un second onglet Termux (glisser depuis la gauche → Nouvelle session) :

```
openclaw health --json
openclaw doctor
```

i Termux supporte plusieurs sessions simultanées. Gardez le Gateway dans une session dédiée, et utilisez une autre pour les commandes de diagnostic.

5.3 Garder le Gateway actif en arrière-plan

Android peut tuer Termux en arrière-plan. Pour éviter cela :

- Notification persistante Termux : dans les paramètres Android → Applications → Termux → désactiver l'optimisation de batterie
- Ou utiliser tmux/screen pour détacher la session :

```
pkg install tmux -y
tmux new -s gateway
openclaw gateway start
# Ctrl+B puis D pour détacher
```

6. Sécurisation initiale

⚠ Cette étape est obligatoire avant tout usage réel. Un Gateway non durci sur Android est exposé à toute app installée sur le même appareil.

6.1 Audit de posture

```
openclaw security audit --deep
```

Traiter toutes les découvertes critiques avant de continuer. Appliquer les corrections automatiques :

```
openclaw security audit --fix
```

6.2 Politique d'outils minimale

Ajouter dans openclaw.json (section tools) :

```
tools: {
  profile: "messaging",
  deny: [
    "group:automation", "group:runtime", "group:fs",
    "sessions_spawn", "sessions_send", "gateway", "cron"
  ],
  fs: { workspaceOnly: true },
  exec: { security: "deny" },
  elevated: { enabled: false },
},
```

6.3 Checklist de validation

Contrôle	Commande de vérification	Résultat attendu
bind = loopback	openclaw security audit	Aucune alerte d'exposition réseau
Auth configurée	openclaw health --json	auth.mode != none
Permissions disque	ls -la ~/.openclaw/	700 / 600
Heartbeat = 0m	cat ~/.openclaw/openclaw.json grep every	"0m"
Outils restreints	openclaw sandbox explain	Profile messaging actif
Audit propre	openclaw security audit --deep	0 finding critique

7. Workspace et personnalisation

7.1 Initialiser le workspace

```
mkdir -p ~/.openclaw/workspace
cd ~/.openclaw/workspace
git init # si git est installé
```

Créer les fichiers de base :

```
touch AGENTS.md SOUL.md USER.md IDENTITY.md MEMORY.md TOOLS.md HEARTBEAT.md
```

7.2 Répartition des fichiers

Fichier	Rôle	Priorité
AGENTS.md	Consignes opératoires + garde-fous doux	Critique
SOUL.md	Persona, ton, limites de comportement	Haute
IDENTITY.md	Nom, rôle métier, style affiché	Moyenne
USER.md	Profil utilisateur, préférences (sans secrets)	Moyenne
MEMORY.md	Mémoire longue — décisions et conventions	Moyenne
TOOLS.md	Conventions d'usage des outils	Haute
HEARTBEAT.md	Tâches proactives (activer en dernier)	Basse

i Se référer au TP 2 (Personnalisation & Autonomisation) pour le contenu détaillé de chaque fichier. Le workspace Android suit exactement la même logique que sur desktop.

7.3 Versionner le workspace

```
cd ~/.openclaw/workspace
git add .
git commit -m "init: workspace OpenClaw Android"
```

Pour sauvegarder vers un dépôt distant (recommandé) :

```
git remote add origin <url-depot-prive>
git push -u origin main
```

8. Connexion canal WhatsApp (cas d'usage courant)

8.1 Configuration du canal

Ajouter dans openclaw.json :

```
{
  "channels": {
    "whatsapp": {
      "dmPolicy": "pairing",
      "allowFrom": ["+225XXXXXXXXXX"], // votre numéro E.164
      "groups": { "*": { requireMention: true } },
    },
  },
}
```

⚠ Sur Android, WhatsApp et OpenClaw coexistent sur le même appareil. L'approche recommandée est d'utiliser un compte WhatsApp Business dédié à l'agent, distinct du compte personnel.

8.2 Appairage

5. Redémarrer le Gateway : `openclaw gateway restart`
6. Scanner le QR code affiché dans Termux depuis WhatsApp (Appareils connectés)
7. Vérifier l'appairage : `openclaw pairing list whatsapp`
8. Approuver votre numéro : `openclaw pairing approve whatsapp <code>`

8.3 Test de connexion

```
openclaw status --all
```

Le canal WhatsApp doit apparaître comme connecté. Envoyer un message test depuis WhatsApp.

9. Gestion du cycle de vie

9.1 Commandes essentielles

Action	Commande
Démarrer le Gateway	<code>openclaw gateway start</code>
Arrêter le Gateway	<code>openclaw gateway stop</code>
Redémarrer	<code>openclaw gateway restart</code>
État complet	<code>openclaw status --all</code>
Logs en direct	<code>tail -f /tmp/openclaw/openclaw-\$(date +%Y-%m-%d).log</code>
Sauvegarde	<code>openclaw backup create</code>
Audit de sécurité	<code>openclaw security audit --deep</code>
Santé (JSON)	<code>openclaw health --json</code>

9.2 Mise à jour d'OpenClaw

```
npm update -g openclaw
openclaw doctor          # vérifier les migrations de config
openclaw security audit  # re-valider la posture
```

9.3 Sauvegarde

```
openclaw backup create
```

L'archive créée contient config, workspace et état. La stocker hors de l'appareil (cloud privé, NAS).

10. Réponse à incident

⚠ En cas de comportement anormal de l'agent ou de suspicion de compromission, appliquer la procédure ci-dessous dans l'ordre.

10.1 Containment

9. Arrêter le Gateway : `openclaw gateway stop`
10. Passer le canal en mode restreint dans `openclaw.json` : `dmPolicy: "disabled"`
11. Redémarrer le Gateway en mode fermé

10.2 Rotation des secrets

- Jeton Gateway : régénérer avec `openclaw doctor --generate-gateway-token` puis mettre à jour `openclaw.json`
- Clés providers (WhatsApp, modèle IA) : révoquer depuis les dashboards respectifs
- Redémarrer le Gateway après rotation

10.3 Audit post-incident

```
openclaw security audit --deep
```

Consulter les logs et transcripts :

```
ls /tmp/openclaw/  
ls ~/.openclaw/agents/main/sessions/
```

Annexe — Aide-mémoire Android

Chemins importants sur Android / Termux

Élément	Chemin
Configuration Gateway	<code>~/.openclaw/openclaw.json</code>
Workspace agent	<code>~/.openclaw/workspace/</code>
Sessions / transcripts	<code>~/.openclaw/agents/main/sessions/</code>
Logs Gateway	<code>/tmp/openclaw/openclaw-YYYY-MM-DD.log</code>

Élément	Chemin
Approbations exec	~/openclaw/exec-approvals.json
Stockage Android partagé	~/storage/
Prefix npm global	~/npm-global/bin/

Variables d'environnement à ajouter dans ~/.bashrc

```
export PATH=~/npm-global/bin:$PATH
export OPENCLAW_GATEWAY_PASSWORD="<JETON>" # alternative au token en config
```

i Rappel transversal : les garde-fous doux (fichiers .md du workspace) orientent le comportement mais ne le garantissent pas. Seule la configuration openclaw.json (garde-fous durs) constitue une frontière de sécurité réelle.