FR

Guide Pratique

OpenClaw + DeepSeek :
LLM local, open-source et couts API

Strategie hybride, economies reelles et retour terrain BOTUM

Mars 2026

Sommaire

DeepSeek a change la donne : un modele open-source qui rivalise avec GPT-4 sur de nombreux benchmarks, tournant en local via Ollama. Voici comment l'integrer a OpenClaw pour reduire vos couts API de plus de 88%.

1. Pourquoi DeepSeek ?

Performance benchmark

DeepSeek-R1 et DeepSeek-V3 se positionnent parmi les meilleurs modeles open-source de 2025. Sur les taches de raisonnement, code et synthese, ils rivalisent directement avec GPT-4o et Claude Sonnet tout en etant gratuits a l'usage local.

• MMLU : 88,5% (DeepSeek-V3) vs 88,7% (GPT-4o) — quasi-equivalents

• HumanEval (code) : 89,1% — superieur a la moyenne des modeles frontier

• Contexte : jusqu'a 128k tokens sur DeepSeek-V3

• Vitesse inference GPU : 40-80 tokens/sec sur RTX 3090/4090

Open-source et Ollama-compatible

DeepSeek est publie sous licence MIT. Les poids sont disponibles publiquement sur Hugging Face. Ollama — le runtime local le plus populaire — distribue des versions quantisees (Q4, Q5, Q8) pret-a-l'emploi.

```
ollama pull deepseek-r1:7b
```

```
ollama pull deepseek-r1:14b
```

```
ollama pull deepseek-v3
```

■ *La version 7b tourne confortablement sur une machine avec 16 Go de RAM (CPU). La version 14b necessite un GPU avec 16 Go de VRAM pour une latence acceptable.*

> *“L'open-source ne signifie pas compromis sur la qualite. En 2025, le gap avec les modeles proprietaires s'est refermé pour la majorite des cas d'usage enterprise.”*

2. Architecture OpenClaw + Ollama + DeepSeek

Installation

L'integration repose sur trois composants : Ollama comme runtime d'inference local, DeepSeek comme modele, et OpenClaw avec un provider configure pour pointer vers Ollama.

```
# 1. Installer Ollama
```

```
curl -fsSL https://ollama.ai/install.sh | sh
```

```
# 2. Telecharger DeepSeek
```

```
ollama pull deepseek-r1:14b
```

```
# 3. Configurer OpenClaw (config.yml)
```

```
default_model: deepseek-r1:14b
```

```
providers:
```

```
ollama:
```

```
base_url: http://localhost:11434
```

```
models: [deepseek-r1:7b, deepseek-r1:14b, deepseek-v3]
```

Latences observees

Les latences varient selon le materiel. Voici les mesures terrain BOTUM sur les cas d'usage courants (reponse premiere token) :

Materiel	Modele	Latence 1er token	Tokens/sec
RTX 4090 (24 Go)	DeepSeek-R1 14b	0.8 sec	65 t/s
RTX 3090 (24 Go)	DeepSeek-R1 14b	1.2 sec	42 t/s
CPU 32 Go RAM	DeepSeek-R1 7b	4.5 sec	12 t/s
Cloud (API OpenAI)	GPT-4o	0.6 sec	~80 t/s

■ *Pour les taches interactives (reponse rapide a un utilisateur Telegram), la latence CPU de 4.5 sec reste acceptable. Pour les workflows batch, elle est negligeable.*

3. Strategie de routing hybride

La cle n'est pas de remplacer l'API cloud par le local — c'est de router intelligemment selon la nature de la tache. OpenClaw supporte le multi-provider : chaque appel peut cibler un modele different.

5 criteres de decision

Complexite du raisonnement : Logique complexe, code avance, analyse multi-etapes → API cloud (GPT-4o, Claude Sonnet)

Sensibilite des donnees : Donnees confidentielles, PII, secrets business → DeepSeek local (rien ne sort)

Volume de la tache : Traitement batch, resume de logs, transformations repetitives → DeepSeek local

Latence requise : Reponse interactive temps-reel → cloud. Workflow background → local

Budget disponible : En fin de mois, basculer plus de taches vers local pour respecter le budget

Configuration routing dans OpenClaw

```
# Skill avec modele specifique
# Dans le prompt systeme ou config skill :
model: deepseek-r1:14b # force local pour ce skill

# Via le chat (override ponctuel) :
/model deepseek-r1:14b
```

> *“Le routing hybride est une posture, pas une configuration figee. Revoyez-le mensuellement selon l'evolution de votre usage et des modeles disponibles.”*

4. Economies reelles et calcul ROI

Baseline : cout API cloud seul

Avant integration DeepSeek, un usage BOTUM typique sur OpenClaw generait environ :

- ~2 millions de tokens/mois (input + output combines)

• Mix GPT-4o (~70%) + Claude Sonnet (~30%)

• Cout mensuel moyen : ~105 $/mois USD

Apres routing hybride DeepSeek

En basculant ~80% des taches vers DeepSeek local (les taches repetitives, les summaries, la recherche web, les digests) :

• Taches locales (DeepSeek) : ~80% du volume → 0 $

• Taches cloud residuelles : ~20% du volume → ~12,60 $/mois

• Reduction totale : -88% sur les couts API

• Economie mensuelle : ~92,40 $/mois

Break-even GPU

Si vous n'avez pas de GPU et envisagez l'achat :

• RTX 3090 d'occasion : ~600-800 $ CAD

• Electricite supplementaire : ~8-12 $/mois

• Break-even : environ 5 mois avec les economies realisees

• Apres break-even : economie nette de ~80-85 $/mois indefiniment

■ *Ces chiffres sont bases sur l'usage BOTUM en production (Mars 2026). Votre ROI variera selon votre volume et mix de taches. Sur CPU seul (sans GPU), le ROI est immediat — juste plus lent.*

5. Limites a connaitre

Dependance materielle

Sans GPU dedie, les performances CPU restent limitees. DeepSeek-R1 14b sur CPU met 4-8 secondes pour la premiere token — acceptable pour les batchs, difficile pour l'interactif.

Qualite sur taches complexes

Sur le raisonnement multi-etapes complexe, les appels d'outils enchaines, ou la generation de code avance, les modeles frontier (GPT-4o, Claude Sonnet) gardent un avantage mesurable. Ne migrez pas ces taches vers local.

Contexte long

DeepSeek-V3 supporte 128k tokens en theorie, mais en pratique sur Ollama, la degradation de qualite apparait apres ~32k tokens selon le materiel et la version quantisee utilisee.

Maintenance

Contrairement a l'API cloud, le local requiert de la maintenance : mises a jour Ollama, nouvelles versions modeles, gestion de l'espace disque (les modeles font 8-40 Go selon la taille).

> *“Le local n'est pas 'set and forget'. C'est un outil qui demande un operateur. Si vous n'avez pas le temps de maintenir, gardez le cloud pour les taches critiques.”*

6. Retour terrain BOTUM

Ce qu'on utilise en production

• DeepSeek-R1 14b pour : digests email, summaries de logs, recherche web, drafts initiaux

• Claude Sonnet pour : raisonnement complexe, code production, decisions importantes

• GPT-4o pour : taches multimodales, generation de contenu blog

• Mix actuel : ~75% local / 25% cloud

Ce qu'on referait

• Commencer par DeepSeek-R1 7b pour tester sans GPU — les resultats sont deja impressionnants

• Mettre en place le routing explicite des le debut — pas de migration a la chaine plus tard

• Monitorer les tokens par provider des le jour 1 — impossible de sauvegarder sans metrics

• Garder une liste des taches 'non-negociables cloud' — evite les regressions de qualite

Ce qu'on ne referait pas

• Migrer toutes les taches d'un coup — tester progressivement, section par section

• Ignorer la latence CPU pour les taches interactives — l'experience utilisateur en souffre

• Sous-estimer la gestion disque — prevoir 100 Go minimum pour plusieurs modeles

7. Checklist deploiement DeepSeek + OpenClaw

- ■ Ollama installe et fonctionne (`ollama serve` demarre au boot)
- ■ DeepSeek-R1 14b (ou 7b selon votre materiel) telecharge et teste
- ■ OpenClaw configure avec provider Ollama (base_url localhost:11434)
- ■ Test de sanite : `/model deepseek-r1:14b` puis une question simple
- ■ Routing hybride defini : liste des taches locales vs cloud
- ■ Monitoring tokens par provider en place
- ■ Budget mensuel API cloud redefini (objectif -80% minimum)
- ■ Alertes sur depassement de budget cloud configures
- ■ Procedure de mise a jour Ollama documentee
- ■ Gestion espace disque : cron de nettoyage si necessite
- ■ Tests de regression : verifier que la qualite reste acceptable sur les taches basculees
- ■ Documentation interne : qui route quoi, pourquoi, depuis quand

Conclusion

DeepSeek + Ollama + OpenClaw forme une combinaison puissante pour les equipes qui veulent reprendre le controle de leurs couts IA sans sacrifier la qualite. Le routing hybride est la cle : ni tout local, ni tout cloud, mais le bon modele pour la bonne tache.

Avec -88% sur les couts API et un break-even GPU de 5 mois, l'investissement se justifie rapidement. Et pour les donnees sensibles, le benefice de ne pas envoyer de donnees hors de votre infrastructure est incalculable.

Billet 8 : OpenClaw et les workflows multi-agents — comment orchestrer plusieurs agents specialises.

Article complet : blog.botum.ca/openclaw-deepseek-llm-local-open-source-couts-api

Site web : www.botum.ca • contact@botum.ca