

Configurer un coffre partagé avec son équipe

Créer un repo GitHub partagé pour les notes d'équipe, le lier à ton vault avec un symlink, et onboarder tes collègues.

Source : <https://prisme.one/plateforme/ia-juillet-2026/semaine-1-coffre-equipe>

Plusieurs projets, plusieurs coffres, et travailler en équipe

LOOM

Plusieurs projets, plusieurs coffres, et travailler en équipe

<https://www.loom.com/share/7e59489752e04a0989e0c384f31ca658>

Travailler sur plusieurs projets en parallèle

Ton second cerveau est parfait pour ton usage perso. Mais très vite, tu vas vouloir utiliser ton IA sur d'autres choses : un site web, une app, un projet client, des notes d'équipe. La question devient : comment ouvrir l'IA sur plusieurs projets sans tout mélanger ?

Il y a trois options. Tu choisis selon ton besoin.

Option A : un projet OpenCode par projet (cloisonnement)

Tu ouvres OpenCode Desktop directement sur le dossier d'un projet. C'est un **projet OpenCode** à part entière : l'IA ne voit que ce dossier, rien d'autre.

Option A : un projet OpenCode par dossier

Second Cerveau/ ← projet OpenCode si tu bosses sur tes notes

opencode.json

AGENTS.md

0 INBOX/, 1 PROJETS/, 2 CASQUETTES/...

mon-site-web/ ← projet OpenCode si tu bosses sur ton site

opencode.json

AGENTS.md

src/, components/...

notes-equipe/ ← projet OpenCode si tu bosses avec l'équipe

opencode.json

AGENTS.md

Pour qui : projets clients sensibles, contextes vraiment différents, ou notes d'équipe que tu ne veux pas mélanger à ton perso. Si tu utilises Obsidian, chaque dossier est un coffre Obsidian séparé. Tu bascules avec « Switch vault ».

La seule contrainte technique : le fichier `opencode.json` doit être **à la racine de chaque projet**, là où tu lances l'IA. C'est lui qui dit où trouver les skills, les MCP, la config du projet.

```
{
  "$schema": "https://opencode.ai/config.json",
  "skills": {
    "paths": [". /4 TOOLS/skills"]
  }
}
```

L'AGENTS.md à la racine décrit le contexte spécifique du projet. Comme ça l'IA a son cadre dès qu'elle démarre.

Option B : un Workspace racine, plusieurs dossiers dedans (recommandé par défaut)

Tu ouvres toujours OpenCode au même endroit : un dossier `Workspace` racine. C'est un **seul projet OpenCode** qui contient tous tes sous-dossiers (second cerveau + projets web + projets clients...).

Option B : un seul projet OpenCode racine

Workspace/ ← projet OpenCode, toujours ici

AGENTS.md ← décrit chaque dossier

opencode.json ← skills.paths, MCP, config globale

Second Cerveau/ ← ouvert dans Obsidian

0 INBOX/, 1 PROJETS/, 2 CASQUETTES/...

mon-site-web/ ← repo perso
projet-client-A/ ← repo client
projet-perso/ ← side project

Pour qui : la plupart des cas. Tu n'as jamais à te demander où ouvrir l'IA. C'est toujours Workspace.

Seule chose à faire : quand tu ajoutes un nouveau dossier, mets à jour le `AGENTS.md` à la racine pour expliquer ce qu'il contient. L'IA lit ce fichier au démarrage et sait quel dossier consulter selon ta question.

Exemple d'extrait à mettre dans `Workspace/AGENTS.md` :

```
## Arbre du Workspace

- `Second Cerveau/` : mon coffre Obsidian (notes perso, projets, casquettes)
- `mon-site-web/` : repo du site Astro déployé sur Cloudflare Pages
- `projet-client-A/` : notes et scripts pour le client A, sensible, ne pas pousser de secrets
- `projet-perso/` : side project libre, expérimentations
```

Le fichier `opencode.json` à la racine, lui, sert à exposer les skills et les MCP à toutes tes sessions :

```
{
  "$schema": "https://opencode.ai/config.json",
  "skills": {
    "paths": [". / Second Cerveau / 4 TOOLS / skills"]
  }
}
```

Option C : pas migrer du tout, brancher l'IA sur Notion ou Outline (MCP)

Si ton équipe utilise déjà **Notion**, **Outline**, **Confluence** ou un autre outil de doc interne, **ne migre pas tout dans Obsidian**. Tu peux brancher l'IA directement sur la base existante via

un **MCP** (Model Context Protocol).

Option C : OpenCode branché sur la base d'équipe via MCP

Toi

Second Cerveau Obsidian local

↔ MCP ↔

Ton équipe

Notion / Outline / Confluence

Tes collègues continuent leur outil, toi tu as ton coffre Obsidian + accès en lecture/écriture à leur base via l'IA.

Le scénario : tu installes le MCP Notion dans ton opencode . json. Tu donnes tes credentials. À partir de là, l'IA peut **lire et écrire** dans la base Notion de l'entreprise. Tes collègues continuent d'utiliser Notion comme avant, toi tu gardes ta vélocité IA sur ton coffre Obsidian local plus l'accès via MCP.

Pour qui : équipes non techniques ou organisations établies qui ont déjà des centaines de pages dans Notion/Outline. Migrer tout dans Obsidian serait un projet de plusieurs mois pour zéro gain. Le MCP te permet d'avoir le meilleur des deux mondes immédiatement.

Comment connecter un MCP : voir le skill /connect-mcp (semaine 3) ou la leçon Installer ses outils IA.

Récap : quelle option pour quel cas

Ton cas	Option
Plusieurs projets indépendants, contextes très différents	Option A : un projet OpenCode par dossier
Tu vis tes journées entre plusieurs dossiers liés (perso + pro + side)	Option B : un Workspace racine, MAJ AGENTS.md
Ton équipe est déjà sur Notion / Outline et ne veut pas bouger	Option C : MCP vers la base existante

Par défaut, démarre avec l'**Option B**. C'est la plus simple à vivre au quotidien. Tu passeras à A ou C uniquement si ton contexte l'exige.

Addon : tout voir dans un seul Obsidian (symlinks et junctions)

Cas spécifique : tu es sur l'**Option A** avec plusieurs coffres Obsidian séparés (par exemple, ton coffre perso + un coffre d'équipe synchronisé via Obsidian Sync). Tu trouves chiant de basculer en permanence entre les deux. Tu voudrais voir **toutes tes notes dans un seul Obsidian**, le perso, avec le coffre d'équipe qui apparaît comme une casquette dedans.

La solution : un **raccourci système** qui pointe d'un dossier à un autre, sans le dupliquer. Sur Mac/Linux on appelle ça un **symlink**. Sur Windows, on utilise une **junction** (`mklink /J`).

Copier le dossier

Deux copies du même contenu

Si tu modifies l'une, l'autre ne change pas

Prend le double de place

Impossible à synchroniser proprement

Symlink ou junction

Un seul exemplaire physique

Modification visible partout

Zéro duplication

La sync ne tourne que sur le vrai dossier

Un seul Obsidian, plusieurs dossiers liés

Workspace/

Second Cerveau/ ← celui que tu ouvres dans Obsidian

0 INBOX/, 1 PROJETS/

2 CASQUETTES/

Coaching/, Dance/, Voyages/...

Notes-Equipe/ → symlink / junction vers ../notes-equipe

notes-equipe/ ← le vrai dossier (coffre d'équipe synchronisé à part)

Process/, Réunions/, Marketing/...

Sur Mac / Linux : symlink (`ln -s`)

```
# Tu as un dossier séparé qui vit ailleurs dans le Workspace
~/Workspace/notes-equipe/
```

```
# Tu crées un symlink dans ton Second Cerveau, dans 2 CASQUETTES
```

```
cd ~/Workspace/Second\ Cerveau/2\ CASQUETTES/  
ln -s ../../notes-equipe Notes-Equipe
```

Sur Windows : junction (`mklink /J`)

Sur Windows, les vrais symlinks demandent le mode développeur ou des droits admin. Pas pratique. **L'équivalent fiable et sans droits admin pour les dossiers locaux**, c'est la **junction**. Obsidian Desktop, OpenCode et Git la traitent exactement comme un dossier normal.

Ouvre un terminal (cmd ou PowerShell), pas besoin de mode admin :

```
cd "C:\Users\Toi\Workspace\Second Cerveau\2 CASQUETTES"  
mklink /J "Notes-Equipe" "..\..\notes-equipe"
```

Le `/J` crée une **junction** (lien de jonction). C'est ce qui se rapproche le plus d'un symlink Unix dans la pratique : Windows considère le dossier comme réel, Obsidian Desktop le suit comme n'importe quel dossier.

À noter : une junction ne marche que pour des dossiers **locaux sur le même volume** (même lettre de lecteur). Si tu pointes vers un dossier sur un autre disque ou sur un partage réseau, utilise plutôt `mklink /D` (symlink vrai) qui exige le mode développeur Windows activé.

Trois points importants

1. **Exclus le dossier lié d'Obsidian Sync** du Second Cerveau qui contient le lien. Sinon Obsidian Sync va synchroniser deux fois le même contenu (une fois dans le coffre d'équipe, une fois dans ton perso via le lien) et créer des conflits. Va dans **Settings → Sync → Selected folders** et désélectionne le dossier symlinké/junction.
2. **Sur mobile (Obsidian iOS / Android), les symlinks et junctions ne sont pas suivis.** Si tu travailles beaucoup sur mobile, oublie cet add-on : reste sur l'Option A et bascule entre coffres avec « Switch vault », ou passe à l'Option B où tout est nativement dans un seul dossier.
3. **Pour partager des skills entre coffres**, le symlink/junction n'est pas la meilleure approche. Utilise plutôt `skills.paths` dans `opencode.json` (voir section dédiée

plus bas).

Synchroniser un coffre avec ton équipe

Une fois la structure choisie, comment plusieurs personnes partagent un même coffre Obsidian ?

Privé : ton coffre perso

Journal, phases de vie

Réflexions intimes

Objectifs perso

Notes de lecture, veille

Contexte global Eliott

Partagé : notes d'équipe

Process de la boîte

Notes de réunion

Marketing, copy

Documentation produit

Clients, coaching

Deux solutions selon ton équipe (la troisième, c'était l'Option C ci-dessus : MCP vers Notion/Outline).

Solution recommandée : Obsidian Sync partagé

C'est la solution la plus simple **si chaque membre de l'équipe peut avoir une licence Obsidian Sync** (4 € par mois et par membre, payé par l'entreprise typiquement).

1. Un membre crée un coffre Obsidian dédié à l'équipe (séparé de son perso).
2. Il active **Obsidian Sync** dessus et invite les autres membres depuis l'interface.
3. Chacun installe Obsidian, se connecte à son compte, sélectionne le coffre partagé, entre le mot de passe d'encryption commun.
4. Tout est synchronisé en temps réel, chiffré de bout en bout (E2E).

Coût : 4 €/membre/mois. Pour une équipe de 5, c'est 20 €/mois. Largement rentable comparé à n'importe quelle autre solution une fois que tu as testé la stabilité.

Avantages :

- Marche aussi sur **mobile** sans config supplémentaire.

- Pas de Git, pas de clone, pas de Personal Access Token à gérer.
- Conflits gérés automatiquement.
- Historique des versions intégré (jusqu'à 1 an selon ton plan).

Côté IA : chaque membre lance OpenCode sur son workspace racine (Option B) avec le coffre partagé dedans, ou directement sur le coffre d'équipe (Option A).

Alternative gratuite : Syncthing

Si tu ne veux pas payer Obsidian Sync, Syncthing fait de la synchronisation peer-to-peer entre les machines des membres. C'est open source, gratuit, plus technique à mettre en place. Pas d'app mobile aussi propre qu'Obsidian Sync (app Android officielle correcte, rien d'officiel sur iOS).

Pour qui : équipes tech qui ont déjà un setup serveur, ou bricoleurs qui acceptent le coût d'entrée technique.

À tester en sandbox avant de t'engager : la résolution de conflits Syncthing est moins fluide qu'Obsidian Sync, et il faut que les machines soient allumées en même temps pour la sync (ou avoir un nœud serveur permanent).

Partager des skills en équipe

Tu as créé un skill custom utile (par exemple `/sync-compta`, `/newsletter`, `/onboarding-client`) et tu veux que toute l'équipe puisse l'utiliser ?

Approche simple : dans le coffre partagé

Si ton équipe partage déjà un coffre via Obsidian Sync, les skills vivent dans Second Cerveau d'équipe/4 TOOLS/skills/ et sont **automatiquement synchronisés** à tout le monde. C'est l'option la plus naturelle. Tu n'as rien d'autre à configurer côté équipe.

Chaque membre référence ce dossier dans son `opencode.json` racine (Option B) :

```
{
  "$schema": "https://opencode.ai/config.json",
  "skills": {
    "paths": [
      "./Second Cerveau/4 TOOLS/skills",
      "./Second Cerveau d'équipe/4 TOOLS/skills"
    ]
  }
}
```



```
]
}
}
```

Approche multi-coffres : `skills.paths` étendu

Si chaque membre a son propre coffre perso, et qu'il y a en plus un dossier de skills d'équipe partagé séparément (Obsidian Sync dédié, repo Git, Syncthing, peu importe), tu pointes vers ces deux dossiers dans `skills.paths` :

```
{
  "$schema": "https://opencode.ai/config.json",
  "skills": {
    "paths": [
      "./Second Cerveau/4 TOOLS/skills",
      "./skills-equipe"
    ]
  }
}
```

OpenCode charge les skills des deux dossiers, et tout fonctionne dans la même interface. Chacun garde ses skills perso (`/done`, `/welcome`, `/weekly-review...`), et tout le monde profite des skills d'équipe (`/onboarding-client`, `/newsletter...`).

skills.paths multi-dossiers

Second Cerveau/4 TOOLS/skills/ ← perso

welcome.md, done.md, weekly-review.md

skills-equipe/ ← partagé (Obsidian Sync ou autre)

onboarding-client.md, newsletter.md

↓ chargés ensemble par `opencode.json`

OpenCode Desktop

`/welcome`, `/done`, `/weekly-review`

`/onboarding-client`, `/newsletter`

Points importants

Toujours un `opencode.json` à la racine du dossier où tu lances l'IA. C'est lui qui dit à OpenCode où trouver les skills, les MCP, la config. Sans ça, les skills ne sont pas chargés et

tu te demandes pourquoi `/welcome` ne fait rien.

AGENTS.md à la racine, c'est le contexte de base. L'IA le lit au démarrage. Si tu changes d'option (passer de A à B par exemple), pense à mettre à jour ce fichier pour qu'il décrive bien la nouvelle structure.

Sur desktop, Obsidian suit nativement les symlinks (Mac/Linux) et les junctions (Windows). Sur mobile, oublie : utilise plutôt l'Option B ou Obsidian Sync sur un coffre dédié.

Obsidian Sync ne synchronise pas les fichiers cachés (`.opencode/`, `.claude/`, `.git/`). C'est pour ça qu'on met les skills dans `4 TOOLS/skills/` (visible), pas dans `.opencode/` (caché). Vrai aussi pour l'AGENTS.md du Workspace : il doit être à la racine, pas dans un dossier caché.

Checklist

- ☐ Je connais les 3 options : A (un projet OpenCode par dossier), B (un Workspace racine), C (MCP vers Notion/Outline)
- ☐ Je sais que par défaut, je pars sur l'Option B : un Workspace racine avec tous mes dossiers dedans
- ☐ Je sais que `opencode.json` doit être à la racine du dossier où je lance l'IA
- ☐ Je sais que AGENTS.md à la racine décrit chaque sous-dossier pour orienter l'IA
- ☐ Pour partager un coffre Obsidian en équipe, Obsidian Sync à 4 € par membre est la solution la plus simple
- ☐ Je connais Syncthing comme alternative gratuite, et le MCP Notion/Outline si l'équipe ne veut pas migrer
- ☐ Je sais que `skills.paths` dans `opencode.json` accepte plusieurs dossiers de skills
- ☐ Si je veux tout voir dans un seul Obsidian : symlink (Mac/Linux) ou junction mklink /J (Windows), et exclure le dossier d'Obsidian Sync

Vidéos

LOOM

Plusieurs projets, plusieurs coffres, et travailler en équipe

<https://www.loom.com/share/7e59489752e04a0989e0c384f31ca658>

Extraction verbatim depuis Prisme One · <https://prisme.one/plateforme/ia-juillet-2026/semaine-1-coffre-equipe>