cmux *mémo*

Commandes et raccourcis essentiels de **cmux**. Voir aussi **les avis sur cmux**.

Famille Postes de pilotage multi-agent · **Éditeur** Manaflow · **Licence** GPL-3.0-or-later
Note du wiki 11/15 · **Commandes vérifiées le** 10 octobre 2026
Fiche, notes et avis aitoolswiki.viellard.fr/outils/postes-de-pilotage/cmux

Installation

```
brew tap manaflow-ai/cmux
brew install --cask cmux
```

`sudo ln -sf "/Applications/cmux.app/Contents/Resources/bin/cmux" /usr/local/bin/cmux`	Lier manuellement la CLI si nécessaire

```
cmux list-workspaces
```

Prérequis : macOS 14 ou version ultérieure. La TUI distincte couvre aussi Linux et Windows.

Notifications

```
cmux notify --title "Tests" --body "La suite est terminée"
cmux notify --title "Agent" --subtitle "fix-login" --body "Validation requise"
cmux list-notifications
cmux list-notifications --json
cmux clear-notifications
```

Raccourci	Action
⌘⇧I	Ouvrir le panneau de notifications.

Notifications (suite)

Raccourci	Action
⌘⇧U	Aller à la notification non lue la plus récente.

Métadonnées dans la barre latérale

`cmux set-status build "compilation" --icon hammer --color "#ff9500" --priority 80`	Afficher un état nommé et sa priorité
`cmux sidebar-state --workspace workspace:2`	Lire l'état d'un workspace

Utiliser une clé stable comme **`build`**, **`tests`**, **`deploy`** ou **`review`** pour que chaque script mette à jour son propre badge.

Automatisation de base

La CLI permet notamment :

```
list-workspaces   new-workspace   new-split
send              send-key        read-screen
open-url          tree            layout
notify            set-status      sidebar-
state
```

Toujours consulter **`cmux`** *`<commande>`* **`--help`** : l'API évolue rapidement.

Configuration

- Global : **`~/.config/cmux/cmux.json`**
- Projet : **`.cmux/cmux.json`**
- Terminal : reprend thèmes, polices et couleurs de **`~/.config/ghostty/config`**
- Hooks de notifications : configurables dans **`notifications.hooks`**

Organisation recommandée

Workspace	Panes conseillés
Fonctionnalité	Agent principal · tests/watch · navigateur.
Revue	Agent reviewer · diff/terminal · CI.
Incident	Agent diagnostic · logs · navigateur/monitoring.

cmux ne crée pas automatiquement les worktrees. Pour des modifications parallèles du même dépôt :

```
git worktree add ../mon-app-fix -b fix/login
git worktree add ../mon-app-feature -b feature/search
```

Ouvrir ensuite un workspace cmux dans chaque dossier.

Garde-fous

- Une branche et un worktree par changement livrable.
- Un port et, si nécessaire, une base distincte par workspace.
- Ne pas confondre notification « terminée » et changement validé.
- Faire relire le diff avant commit, puis exécuter les tests hors du contexte de l'agent auteur.
- Conserver tmux pour les processus distants persistants si nécessaire.

Sources (documentation officielle)

- Premiers pas
- Référence CLI
- Notifications