

Lab 088: Copilot CLI : l'agent dans votre terminal

Level: L200 Parcours: copilot Durée: ~60 min Coût: GitHub Free

info "Conception gratuite d'abord" Terminez l'exercice avec Copilot Free ou la simulation incluse. Les fonctions payantes sont facultatives et clairement indiquées.

tip "Impact sur les AI Credits" Faible pour l'exercice guidé ; moyen/élevé pour les exécutions agentiques ou répétées. Préférez la sélection automatique du modèle, planifiez avant d'agir, gardez un contexte ciblé et évitez les sessions parallèles inutiles.

Ce que vous apprendrez

Installer et authentifier le Copilot CLI actuel Utiliser les modes interactif et programmatique Configurer agents, hooks, MCP et approbations Appliquer les contrôles de sécurité et de politique

Alignement GH-300

Use Copilot features (25?30%). Cette correspondance suit l'instantané GH-300 du 07/08/2026.

Le plan officiel peut conserver des termes anciens comme « Copilot Edits » ; ce lab enseigne l'expérience actuelle et signale la différence.

Carte conceptuelle

[diagram or code sample]

Exercice pratique

Étape 1: Installer le CLI actuel

Utilisez le package officiel (npm install -g @github/copilot), WinGet, Homebrew ou le binaire. L'ancienne extension gh copilot est retirée.

Étape 2: Commencer en lecture seule

Demandez au CLI d'expliquer le dépôt et de proposer une validation. Examinez le plan et chaque permission.

Étape 3: Ajouter une automatisation contrôlée

Créez un agent de qualité documentaire, un hook bloquant l'affichage de secrets et un outil MCP en lecture seule.

Étape 4: Tester le mode programmatique

Produisez une sortie lisible par machine pour une tâche bornée. Échouez en cas d'écriture non approuvée ou commande destructive.

Kit d'accompagnement

Le build du site génère des téléchargements PDF et ZIP stables. Le ZIP contient le guide du lab et tous les fichiers d'accompagnement déclarés.

Télécharger le PDF généré Télécharger le ZIP généré

Supporting file: .github/agents/cli.agent.md

Plan briefly, make the smallest complete change, run the relevant tests, and summarize evidence.

Supporting file: .github/hooks/pre-command.json

```
{ "hook": "pre-command", "deny": [ "rm -rf", "git reset --hard" ], "require_approval": [ "deploy", "delete" ] }
```

Supporting file: commands.csv

command,purpose,approval copilot -p,bounded prompt,not required copilot --allow-tool,tool execution,required

Supporting file: scripts/session_summary.ps1

```
param([string]$Session = "workshop") Write-Output "Session: $Session"
```