

## Lab 088: Copilot CLI: The Agent in Your Terminal

Level: L200 Journey: copilot Time: ~60 min Cost: GitHub Free

info "Free-first design" Complete the guided exercise with Copilot Free or the included simulation. Paid features are optional and explicitly labeled.

tip "AI Credit impact" Low for guided work; medium/high for agentic or repeated runs.

Prefer automatic model selection, plan before acting, keep context focused, and avoid unnecessary parallel sessions.

### What You'll Learn

Install and authenticate the current Copilot CLI Use interactive and programmatic workflows

Configure agents, hooks, MCP, and approvals Apply command and policy safety checks

GH-300 alignment

Use Copilot features (25?30%). This mapping follows the GH-300 curriculum snapshot dated 2026-08-07. The official outline may retain older product wording such as ?Copilot Edits?; this lab teaches the current product experience and calls out that difference.

### Concept Map

[diagram or code sample]

### Hands-on Exercise

#### Step 1: Install the current CLI

Use the official package (npm install -g @github/copilot), WinGet, Homebrew, or the release binary. The retired gh copilot extension is not this product.

#### Step 2: Start with a read-only prompt

Ask the CLI to explain the repository and propose a validation command. Inspect its plan and every requested permission.

#### Step 3: Add controlled automation

Create a custom agent for documentation quality, a pre-command hook that blocks secret-printing commands, and one read-only MCP tool.

#### Step 4: Exercise programmatic mode

Produce machine-readable output for a bounded task. Fail the workflow if the CLI proposes an unapproved write or destructive command.

### Supporting kit

The site build generates stable PDF and ZIP downloads. The ZIP contains the lab guide and every supporting file declared for this lab.

Download generated PDF Download generated ZIP

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"
```