

START HERE

Copilot AI Credit Lab

A screenshot-by-screenshot practical guide

Created by Akash Dwivedi

Complete a real-looking checkout task in the browser, change six optimization settings, and compare the modeled AI credits. No installation, sign-in, shell, or AI subscription is required for this simulator.

[Open the live lab](#) | [Open the CLI view](#)

The screenshot displays the Copilot AI Credit Lab interface, which is divided into three main sections:

- VS Code Workspace (Left):** Shows a simulated workspace with a file explorer on the left containing a project named 'northstar-commerce'. The main editor displays a TypeScript file 'cart.ts' with code for a shopping cart interface and a 'calculateTotal' function. A 'PROBLEMS' panel at the bottom shows a 'failing fixture assertion' in 'cart.test.ts'.
- AI Credit Inspector (Right):** Displays the 'AI CREDIT INSPECTOR' with a 'YOUR CURRENT REQUEST' section showing a modeled AI credit of 23.80. Below this, a 'Breakdown' table lists various components and their AIC values:

Component	AIC
System	0.56
Instructions	0.96
Tools	2.04
History	5.52
Files	7.44
Your prompt	0.01
Output + reasoning	7.28

The total estimated AI credit use is shown as 23.80 → 23.80 AIC. A 'Run comparison' button is at the bottom.
- Optimization Levers (Bottom):** A section titled 'Your optimization levers' with 6 active levers, each with a toggle switch and a description:
 - Focus the context:** Attach the relevant file, not the entire repository. (Toggle: Off)
 - Trim enabled tools:** Keep file editing and tests. Disable unrelated MCP tools. (Toggle: Off)
 - Refresh the session:** Start a new task in a clean chat, or compact ongoing work. (Toggle: Off)
 - Keep instructions lean:** Use a short, stable project prefix. Cache reuse is conditional. (Toggle: Off)
 - Right-size reasoning:** Reserve high effort for tasks that genuinely need it. (Toggle: Off)
 - Let Auto route:** Model routing can change AIC use without changing raw token count. (Toggle: Off)

The live lab: workspace on the left, AI credit inspector on the right, six optimization switches below.

Choose one route first

First visit: follow the VS Code guided route. **Terminal demonstration:** follow the CLI guided route. Native controls and commands are optional alternatives, not additional steps to perform twice.

Screenshot convention: orange outlines mark the control to click. Screenshots are cropped from the deployed app, not mock-ups. All amounts are fixture estimates, not measured GitHub usage or promised savings.

Live workflow verified: 2026-09-23. Recommended setup: desktop browser at 100% zoom, approximately 1440 pixels wide.

CONTENTS

Find the task you need

Follow one guided route in order, then apply the patch, compare, and export. Each task includes an actual screenshot plus a completion check.

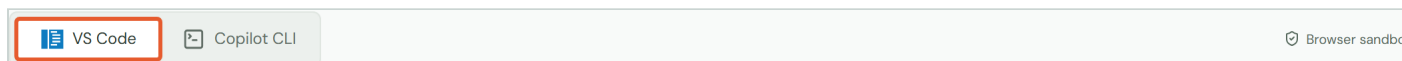
SETUP	3	CLI / TASK 5 OF 10	18
Open a clean checkout demo		Use lean project instructions	
VS CODE / TASK 1 OF 9	4	CLI / TASK 6 OF 10	19
Save the baseline run		Set medium reasoning effort	
VS CODE / TASK 2 OF 9	5	CLI / TASK 7 OF 10	20
Focus on the failing files		Select Auto model routing	
VS CODE / TASK 3 OF 9	6	CLI / TASK 8 OF 10	21
Keep only the relevant tools		Run HydraFusion	
VS CODE / TASK 4 OF 9	7	CLI / TASK 9 OF 10	22
Start a fresh conversation		Recognize the saved optimized run	
VS CODE / TASK 5 OF 9	8	CLI / TASK 10 OF 10	23
Use lean project instructions		Finish the CLI tour	
VS CODE / TASK 6 OF 9	9	VERIFY THE FIX	24
Set medium reasoning effort		Apply the patch and check it	
VS CODE / TASK 7 OF 9	10	COMPARE	25
Select Auto model routing		Read the saved results correctly	
VS CODE / TASK 8 OF 9	11	EXPORT	26
Run the optimized request		Download the run report	
VS CODE / TASK 9 OF 9	12	SHARE	27
Finish without losing your work		Send a reproducible configuration	
CLI / SETUP	13	ADDITIONAL SCENARIO	28
Choose the terminal route		Fix an API retry boundary	
CLI / TASK 1 OF 10	14	ADDITIONAL SCENARIO	29
Save the baseline run		Cover a shipping edge case	
CLI / TASK 2 OF 10	15	UNDERSTAND AIC	30
Focus on the failing files		What the simulator counts	
CLI / TASK 3 OF 10	16	RECOVERY	31
Keep only the relevant tools		Get unstuck without restarting blindly	
CLI / TASK 4 OF 10	17		
Start a fresh conversation			

Open a clean checkout demo

1. Open [the live lab](#). Wait for the code editor to load. A first visit can take longer while the hosted app starts.
2. In **YOUR SCENARIO**, select **Fix a checkout regression**. If the selector is disabled, exit the existing tour with its top-right X first.
3. Click **VS Code** above the workspace. The selected surface stays selected when you start the tour.
4. Click **Guided tour** in the top-right page header. This resets the current demo and starts step **1 / 9**.
5. Keep the default prompt and leave **Assume warm instruction cache** unchecked throughout the main walkthrough.



1. Scenario selector: choose Fix a checkout regression.



2. Workspace tabs: choose VS Code.



3. Top page header: click Guided tour, outlined at the right.

CHECK BEFORE CONTINUING

The inspector reads **23.80 AIC**, all six levers are off, and the tour says **First, establish the baseline.**

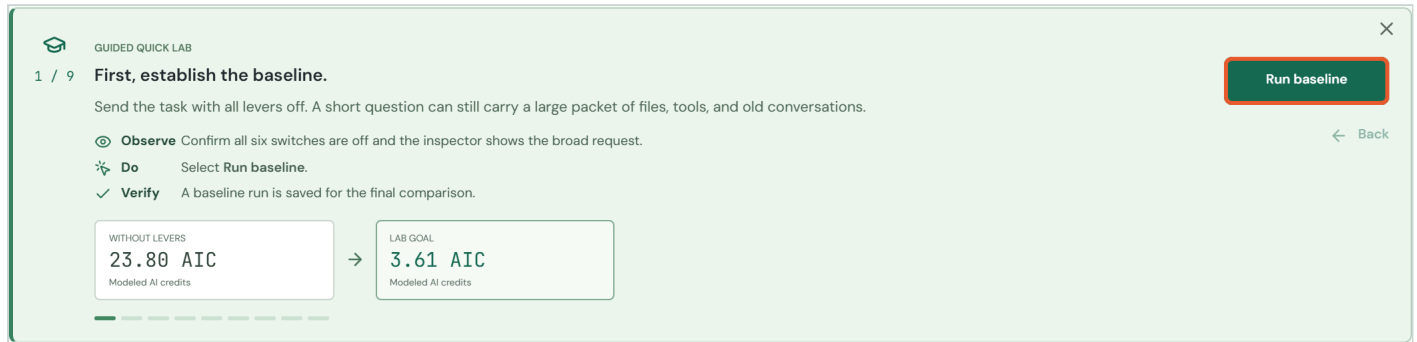
Do not reset between tasks

Continue in the same tab. Starting Guided tour again clears previous runs and code edits. Export anything you need before restarting.

Save the baseline run

Capture the unoptimized request before changing any settings.

1. In the green tour panel above the editor, click **Run baseline** (outlined below).
2. Wait for the action to finish. The same button changes to **Continue**.
3. Check the result below, then click **Continue** once to open the next task.



Guided route: task 1 / 9, before clicking Run baseline.

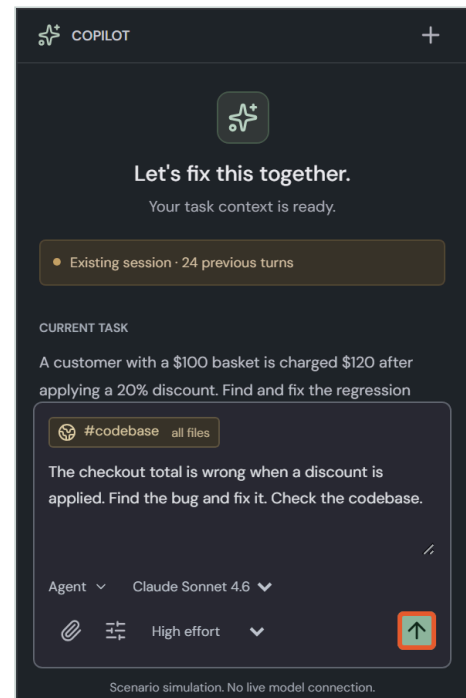
CHECK BEFORE CONTINUING

A simulated diagnosis and reference patch appear. Run history now has one completed run, and the inspector stays at **23.80 AIC**. **Current request: 23.80 → 23.80 modeled AIC**.

Native editor control: Send prompt

Outside the tour, keep the prefilled prompt, then click the upward-arrow **Send prompt** button at the bottom-right of Copilot Chat. Hover to confirm its tooltip. Do not apply the patch yet.

Optional alternative. The tour button already performs this change.



Native-control close-up; orange marks the target.

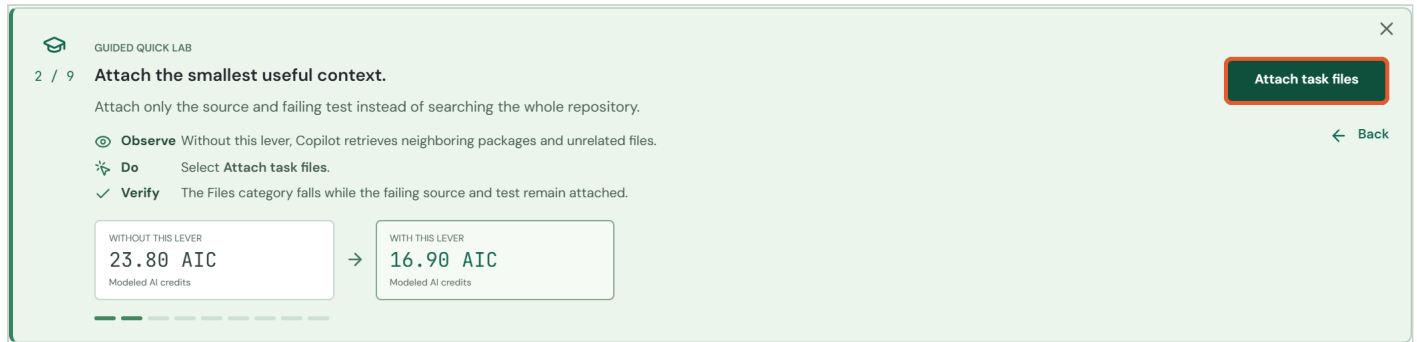
Practical note

The baseline includes broad files, 28 tool schemas, prior conversation history, long instructions, high reasoning effort, and manual model selection.

Focus on the failing files

Keep the source and failing test; remove unrelated repository context.

1. In the green tour panel above the editor, click **Attach task files** (outlined below).
2. Wait for the action to finish. The same button changes to **Continue**.
3. Check the result below, then click **Continue** once to open the next task.



Guided route: task 2 / 9, before clicking Attach task files.

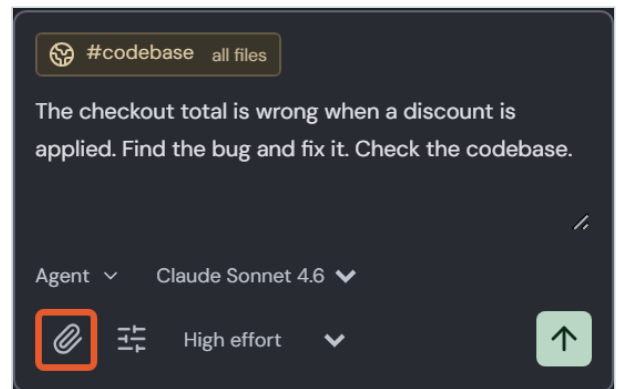
CHECK BEFORE CONTINUING

Focus the context is on. The Files category drops. In Chat, the attachment changes from **#codebase** to **cart.ts +1**. **Current request: 23.80 → 16.90 modeled AIC.**

Native editor control: Attach task files

Click the paperclip at the bottom-left of the Chat composer, tooltip **Attach task files**. The simulator attaches the source and test automatically; there is no file chooser.

Optional alternative. The tour button already performs this change.



Native-control close-up; orange marks the target.

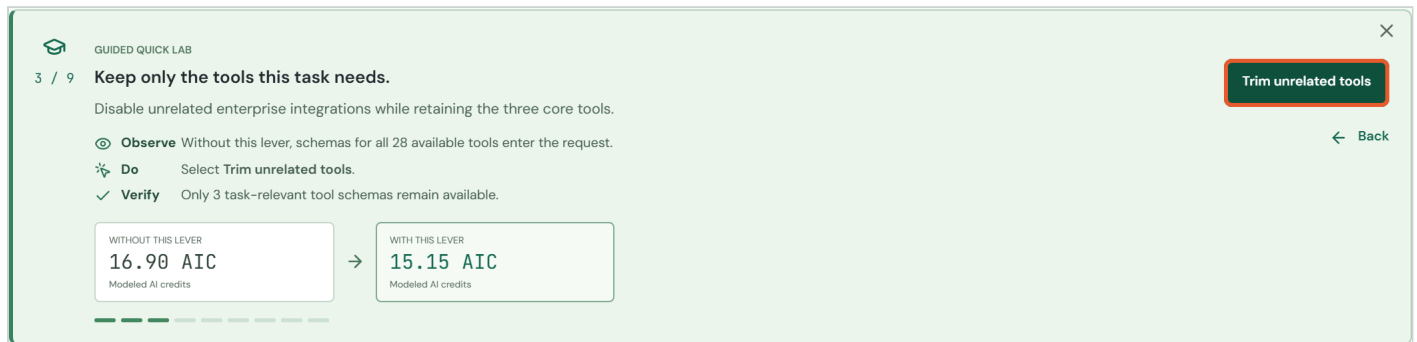
Practical note

Keep the tour prompt unchanged to reproduce the exact AIC values. A custom CLI prompt has a different estimated size and may change the last decimal.

Keep only the relevant tools

Stop including schemas for tools this task does not need.

1. In the green tour panel above the editor, click **Trim unrelated tools** (outlined below).
2. Wait for the action to finish. The same button changes to **Continue**.
3. Check the result below, then click **Continue** once to open the next task.



Guided route: task 3 / 9, before clicking Trim unrelated tools.

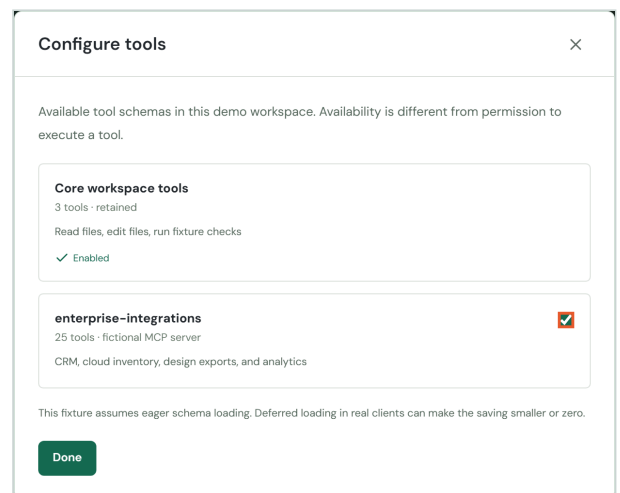
CHECK BEFORE CONTINUING

Trim enabled tools is on. Only the three core workspace tools remain; the fictional 25-tool integration group is disabled. **Current request: 16.90 → 15.15 modeled AIC.**

Native editor control: Configure tools

Click the sliders icon beside the paperclip, tooltip **Configure tools**. In the dialog, **uncheck** the checkbox beside **enterprise-integrations**. Leave Core workspace tools enabled. Click **Done**.

Optional alternative. The tour button already performs this change.



Native-control close-up; orange marks the target.

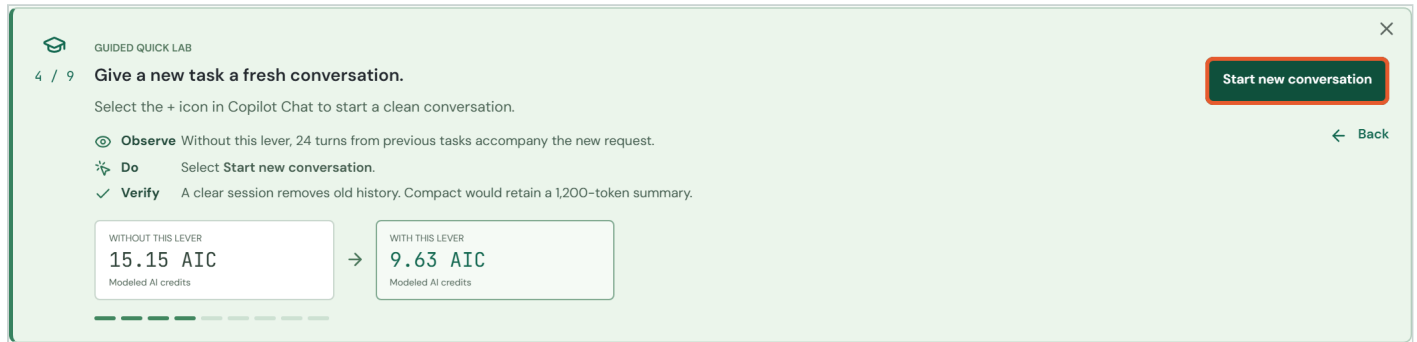
Practical note

enterprise-integrations is fictional. The modeled saving assumes eager schema loading; deferred loading in real clients can make this saving smaller or zero.

Start a fresh conversation

Do not carry unrelated conversation history into a new task.

1. In the green tour panel above the editor, click **Start new conversation** (outlined below).
2. Wait for the action to finish. The same button changes to **Continue**.
3. Check the result below, then click **Continue** once to open the next task.



Guided route: task 4 / 9, before clicking Start new conversation.

CHECK BEFORE CONTINUING

Refresh the session is on. The History category becomes **0.00 AIC**. The old response disappears, but the saved baseline run remains in Run history. **Current request: 15.15 → 9.63 modeled AIC**.

Native editor control: New chat

Click the **+** at the top-right of the Copilot Chat pane, tooltip **New chat**. Do not use the page-level Reset demo button.



Native-control close-up; orange marks the target.

Optional alternative. The tour button already performs this change.

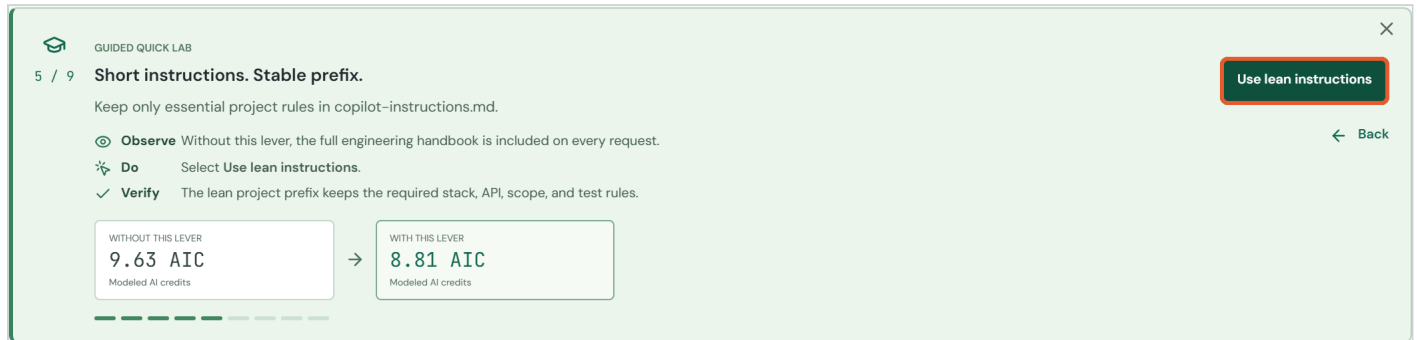
Practical note

For ongoing work, `/compact` retains a 1,200-token summary. It is not equivalent to `/clear` and produces a higher modeled total. The eraser icon only clears the displayed terminal entries.

Use lean project instructions

Retain essential engineering rules without loading a full handbook.

1. In the green tour panel above the editor, click **Use lean instructions** (outlined below).
2. Wait for the action to finish. The same button changes to **Continue**.
3. Check the result below, then click **Continue** once to open the next task.



Guided route: task 5 / 9, before clicking Use lean instructions.

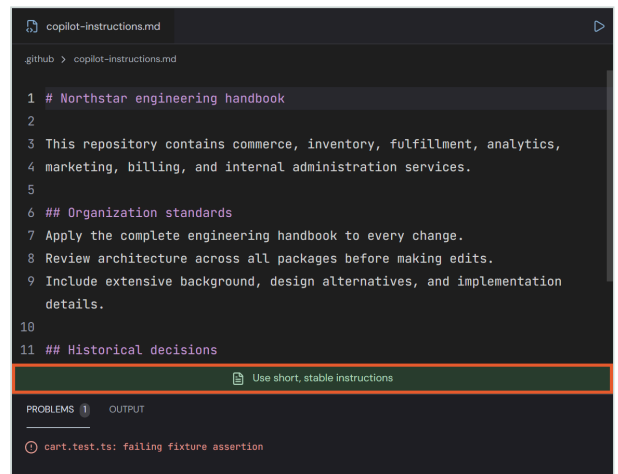
CHECK BEFORE CONTINUING

Keep instructions lean is on. The fixture retains the stack, public-API constraint, focused reads, relevant tests, and minimal patch rules. **Current request: 9.63 → 8.81 modeled AIC.**

Native editor control: instruction file

In the left Explorer, open `.github > copilot-instructions.md`. Click **Use short, stable instructions** below the editor. This fixture is read-only; use the button rather than trying to type into it.

Optional alternative. The tour button already performs this change.



Native-control close-up; orange marks the target.

Practical note

Leave the warm-cache checkbox off for this walkthrough. Shorter instructions do not guarantee a provider cache hit.

Set medium reasoning effort

Match the modeled reasoning budget to this bounded regression.

1. In the green tour panel above the editor, click **Set medium effort** (outlined below).
2. Wait for the action to finish. The same button changes to **Continue**.
3. Check the result below, then click **Continue** once to open the next task.

GUIDED QUICK LAB

6 / 9 **Match reasoning effort to the task.**

This is a bounded regression with an explicit failing assertion, so medium effort is appropriate.

Observe Without this lever, the fixture spends a high reasoning budget exploring alternatives.

Do Select **Set medium effort**.

Verify Medium effort lowers modeled output while preserving the supplied reference fix.

WITHOUT THIS LEVER
8.81 AIC
Modeled AI credits

→

WITH THIS LEVER
4.01 AIC
Modeled AI credits

Set medium effort

← Back

Guided route: task 6 / 9, before clicking Set medium effort.

CHECK BEFORE CONTINUING

Right-size reasoning is on. Output + reasoning decreases. The model may still be Claude Sonnet 4.6 until the next task. **Current request: 8.81 → 4.01 modeled AIC.**

Native editor control: Reasoning effort

Open the **High effort** dropdown in the bottom row of the Chat composer. Select **Medium effort**. Do not change the model dropdown yet.

Optional alternative. The tour button already performs this change.

cart.ts +1

Fix the discount calculation in #file:src/cart.ts. The failing assertion in #file:src/cart.test.ts expects 80, received 120. Keep the public API unchanged.

Agent Claude Sonnet 4.6

High effort

↑

Native-control close-up; orange marks the target.

Practical note

The response is a fixed reference answer. This does not prove that lower reasoning effort preserves quality on arbitrary real tasks.

Select Auto model routing

Demonstrate how a modeled change in pricing affects AIC.

1. In the green tour panel above the editor, click **Select Auto** (outlined below).
2. Wait for the action to finish. The same button changes to **Continue**.
3. Check the result below, then click **Continue** once to open the next task.

GUIDED QUICK LAB

7 / 9 **Let Auto choose an eligible model.**

Select Auto in the model picker. Actual AIC depends on the model selected for each request.

Observe Without this lever, the fixture uses Claude Sonnet 4.6 pricing.

Do Select **Select Auto**.

Verify This demo assumes a 10% lower blended AIC rate for Auto; actual routing and pricing vary.

WITHOUT THIS LEVER
4.01 AIC
Modeled AI credits

→

WITH THIS LEVER
3.61 AIC
Modeled AI credits

Select Auto

← Back

Guided route: task 7 / 9, before clicking Select Auto.

CHECK BEFORE CONTINUING

Let Auto route is on, all **6 / 6** levers are active, and the tour configuration reaches **3.61 AIC**.
Current request: 4.01 → 3.61 modeled AIC.

Native editor control: Chat model

Open the **Claude Sonnet 4.6** dropdown above the Chat composer toolbar. Select **Auto**. Keep Medium effort selected.

Optional alternative. The tour button already performs this change.

cart.ts +1

Fix the discount calculation in #file:src/cart.ts. The failing assertion in #file:src/cart.test.ts expects 80, received 120. Keep the public API unchanged.

Agent ▾ Claude Sonnet 4.6 ▾

Medium effort ▾

Send

Native-control close-up; orange marks the target.

Practical note

The simulator assumes a 10% lower blended rate for Auto. This is not a guaranteed discount or a statement about the model real Auto will select.

Run the optimized request

1. Confirm all six levers are enabled and the inspector reads **3.61 AIC**. Do not change the prompt or enable warm cache.
2. Click **Run optimized task** in the green panel. Wait for the simulated response and patch preview.
3. The primary button changes to **Continue**. Click it once.

The screenshot shows the VS Code interface with a 'GUIDED QUICK LAB' panel. The panel has a progress indicator '8 / 9' and the title 'Same task. Now run it lean.' Below the title, there is a description: 'Rerun the same scenario with all six levers. Compare the packet and reference patch, not just the headline percentage.' There are three sections: 'Observe' (The saved baseline shows how this task looked without optimization.), 'Do' (Select Run optimized task.), and 'Verify' (The optimized run is saved beside the baseline with the same reference outcome.). At the bottom, there are two boxes: 'WITHOUT LEVERS' showing '23.80 AIC' and 'Modeled AI credits', and 'LAB GOAL' showing '3.61 AIC' and 'Modeled AI credits'. A green arrow points from the 'WITHOUT LEVERS' box to the 'LAB GOAL' box. In the top right corner, there is a green button labeled 'Run optimized task' which is highlighted with a green box. There is also a 'Back' button with a left arrow.

Task 8 / 9: Run optimized task.

CHECK BEFORE CONTINUING

Run history contains two completed runs: the original baseline and the optimized request. Both use the same supplied reference fix. Merely toggling switches does not save a run.

Native alternative

Click the Chat composer upward-arrow **Send prompt** button, or the inspector's **Run comparison** button. Use only one, once. Each completed request adds a history row.

The screenshot shows the VS Code Chat composer interface. At the top, there is a tab labeled 'cart.ts +1'. Below the tab, there is a text area containing the prompt: 'Fix the discount calculation in #file:src/cart.ts. The failing assertion in #file:src/cart.test.ts expects 80, received 120. Keep the public API unchanged.' Below the text area, there are several controls: 'Agent' with a dropdown arrow, 'Auto' with a dropdown arrow, and 'Medium effort' with a dropdown arrow. At the bottom right, there is a green button with an upward arrow, which is highlighted with a green box.

Optimized composer with Auto and Medium effort.

Keep the comparison honest

The headline shows about 85% fewer modeled credits. The scoped prompt differs from the original broad prompt; each saved row computes its own all-levers-off baseline for that row's prompt.

Finish without losing your work

- 1. At **Take the habit back to your workspace.**, click **Finish tour**.
- 2. Do not click Guided tour again: it starts over and clears the saved runs.
- 3. Continue to **Apply the patch and check it**, then export your run report before leaving the tab.

 GUIDED QUICK LAB

9 / 9 **Take the habit back to your workspace.**

You have compared two modeled approaches to the same task. Export the run report, then validate gains with your own usage and quality checks.

 **Observe** Review each category rather than treating the headline reduction as a guarantee.

 **Do** Select Finish tour.

 **Verify** Run history now contains a reproducible before-and-after comparison.

WITHOUT LEVERS
23.80 AIC
Modeled AI credits

→

LAB GOAL
3.61 AIC
Modeled AI credits

Finish tour

← Back

Task 9 / 9: Finish tour closes the panel and keeps your results.

 Your demo runs 2

Export report

Run / scenario	Workspace	Levers	Baseline AIC	Configured AIC	AIC reduction
#2 Fix a checkout regression Cold cache · New-chat preset	 CLI	6 / 6	23.80	3.61	85%
#1 Fix a checkout regression Cold cache · New-chat preset	 CLI	0 / 6	23.80	23.80	0%

Each row compares the same scenario and prompt with all levers off versus the captured configuration. AIC estimates are frozen when a run completes and are not billing telemetry.

Example from the CLI guided route: the same two-run structure, with CLI in the Workspace column.

CHECK BEFORE CONTINUING

The guided panel closes. Your lever selections and two completed runs remain. The tour does not automatically apply the reference patch.

Next route is optional

The CLI route begins on the next page. Export the VS Code run report first if you want to preserve it, because starting the CLI tour resets the demo.

Choose the terminal route

1. Open [the CLI link](#), or click **Copilot CLI** above the workspace.
2. Select **Fix a checkout regression** in YOUR SCENARIO.
3. For the guided route, click **Guided tour** in the top-right header. Confirm **1 / 10**.
4. Follow the next ten tasks in order. Use the tour action buttons for the exact published totals and two-run comparison.



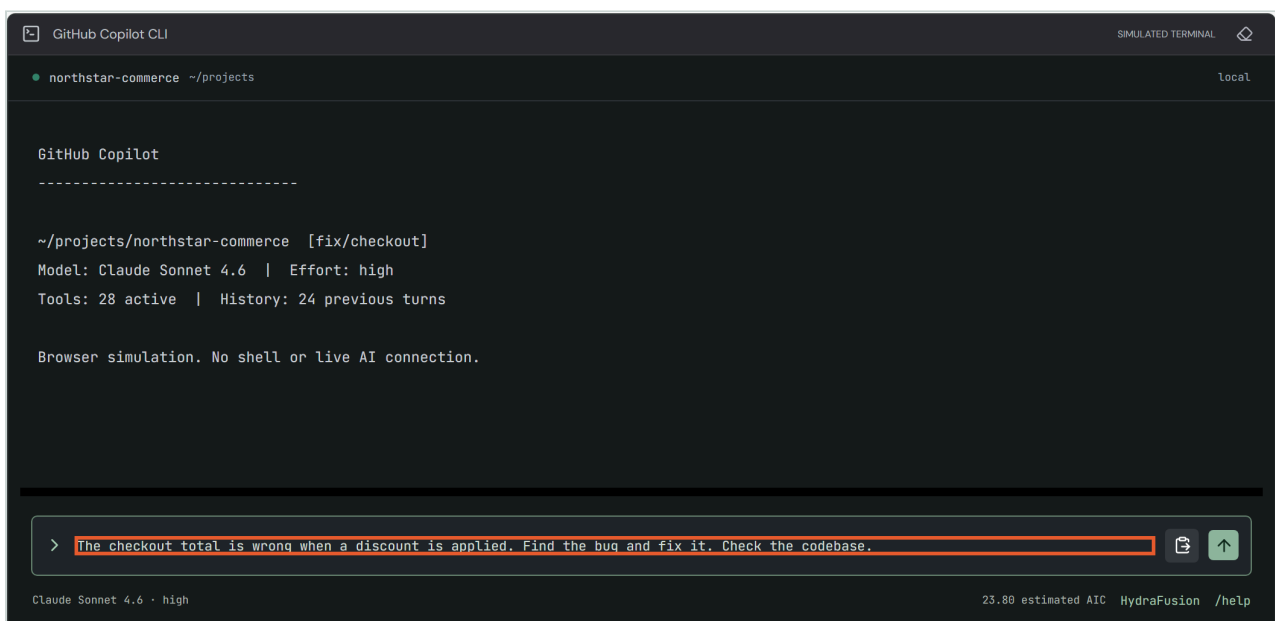
Select Copilot CLI before starting Guided tour.



Click Guided tour in the page header, not the /help button inside the terminal.

Guided route versus command route

Each task below shows the tour button plus an optional manual command. Choose one route. Manual prompts execute immediately and can create additional history rows. Do not run commands as extra steps during the guided route.

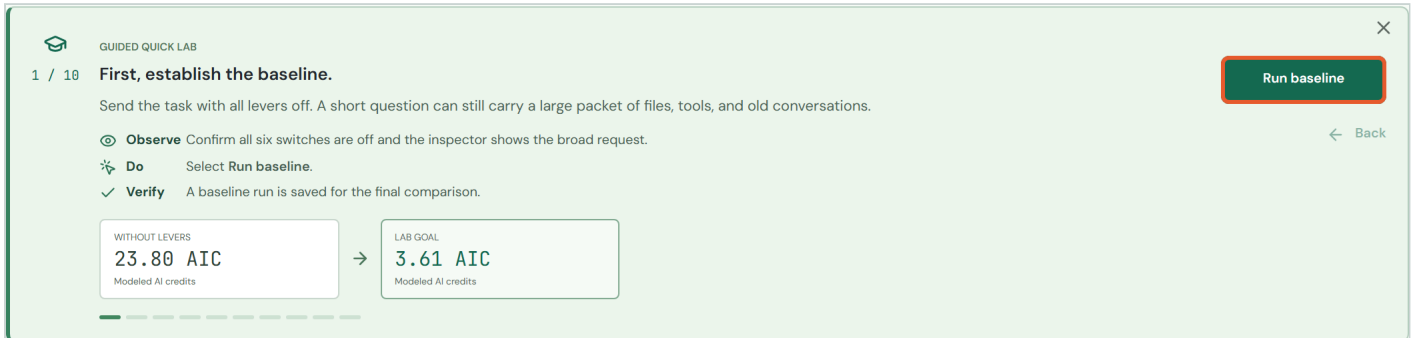


For manual commands, type in the single-line input at the bottom, not in the dark output area.

Press Enter once to submit. Wait until the input is enabled again before the next command. These commands belong inside this browser demo, not PowerShell, Azure CLI, or gh copilot.

Save the baseline run

1. In the green tour panel, click **Run baseline**.
2. Wait for **Continue** to become enabled. Check the result, then click **Continue** once.



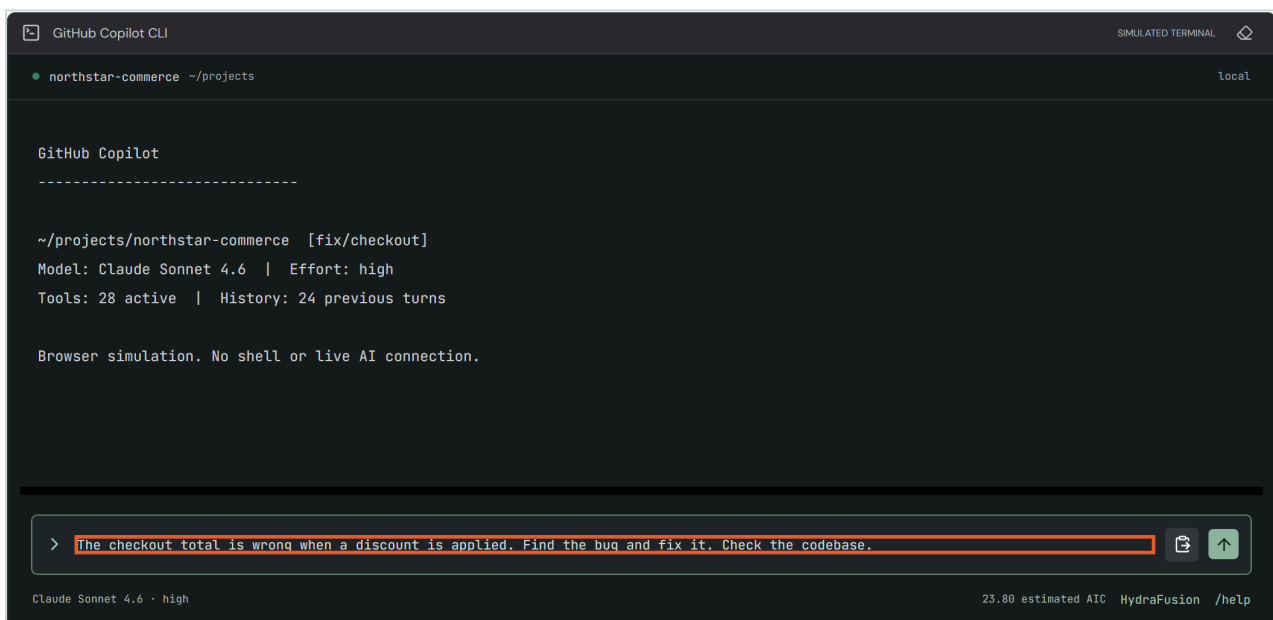
Guided CLI route: task 1 / 10.

CHECK BEFORE CONTINUING

A simulated diagnosis and reference patch appear. Run history now has one completed run, and the inspector stays at **23.80 AIC. Guided value: 23.80 → 23.80 AIC.**

Manual CLI alternative

Outside the tour, click the clipboard icon beside the CLI input (**Insert scenario prompt**), then press **Enter**. Wait until the input is enabled again.

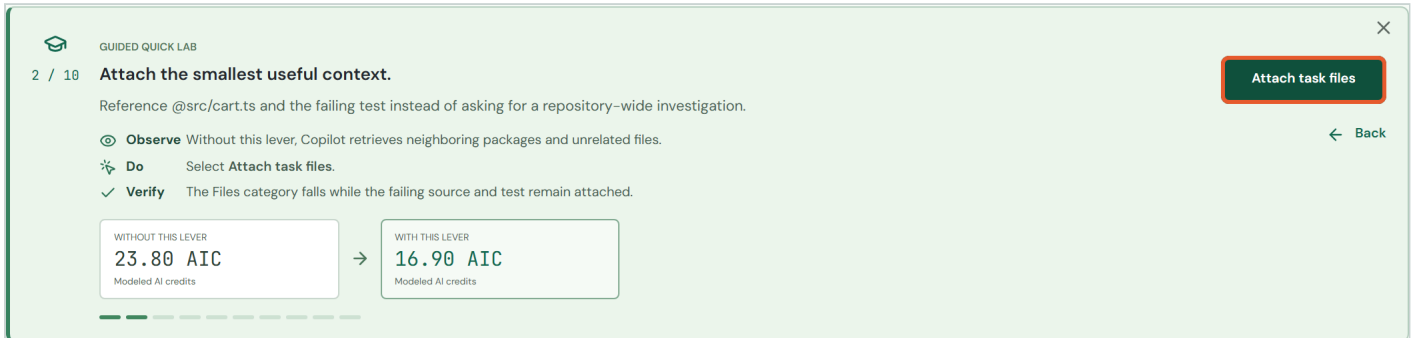


Manual route: the highlighted input or picker is the control to use.

The baseline includes broad files, 28 tool schemas, prior conversation history, long instructions, high reasoning effort, and manual model selection.

Focus on the failing files

1. In the green tour panel, click **Attach task files**.
2. Wait for **Continue** to become enabled. Check the result, then click **Continue** once.



Guided CLI route: task 2 / 10.

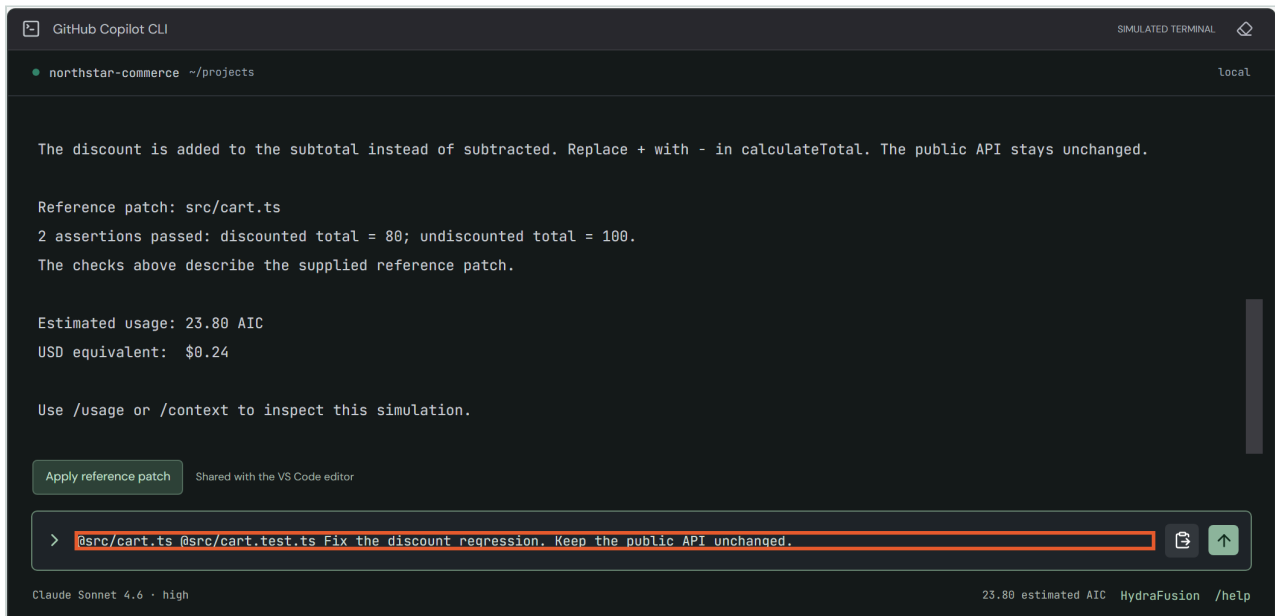
CHECK BEFORE CONTINUING

Focus the context is on. The Files category drops. **Guided value: 23.80 → 16.90 AIC.**

Manual CLI alternative

In the CLI input, type the following and press **Enter**. This both enables focused context and runs the fixture, so the manual route adds a run here.

```
@src/cart.ts @src/cart.test.ts Fix the discount regression. Keep the public API unchanged.
```

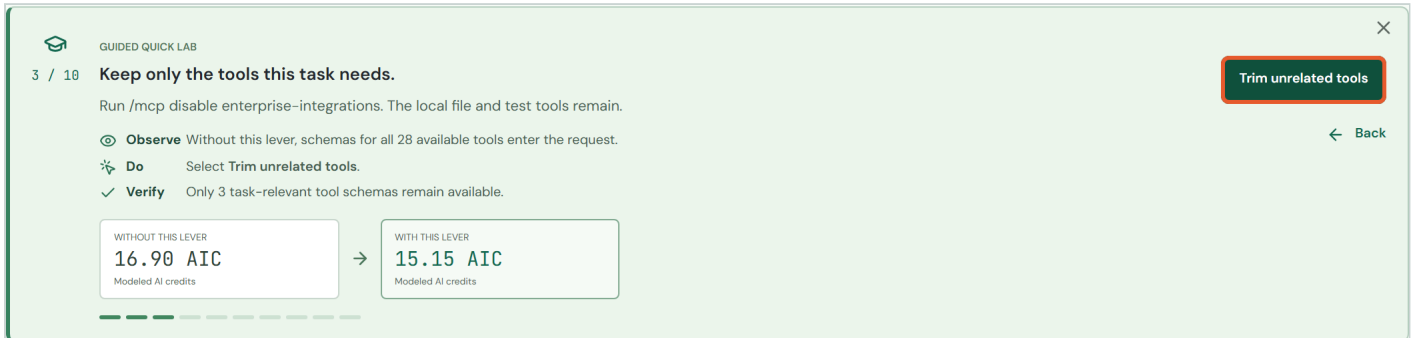


Manual route: the highlighted input or picker is the control to use.

Keep the tour prompt unchanged to reproduce the exact AIC values. A custom CLI prompt has a different estimated size and may change the last decimal.

Keep only the relevant tools

1. In the green tour panel, click **Trim unrelated tools**.
2. Wait for **Continue** to become enabled. Check the result, then click **Continue** once.



GUIDED QUICK LAB

3 / 10 **Keep only the tools this task needs.**

Run `/mcp disable enterprise-integrations`. The local file and test tools remain.

Observe Without this lever, schemas for all 28 available tools enter the request.

Do Select Trim unrelated tools.

Verify Only 3 task-relevant tool schemas remain available.

WITHOUT THIS LEVER
16.90 AIC
Modeled AI credits

→

WITH THIS LEVER
15.15 AIC
Modeled AI credits

Trim unrelated tools

Back

Guided CLI route: task 3 / 10.

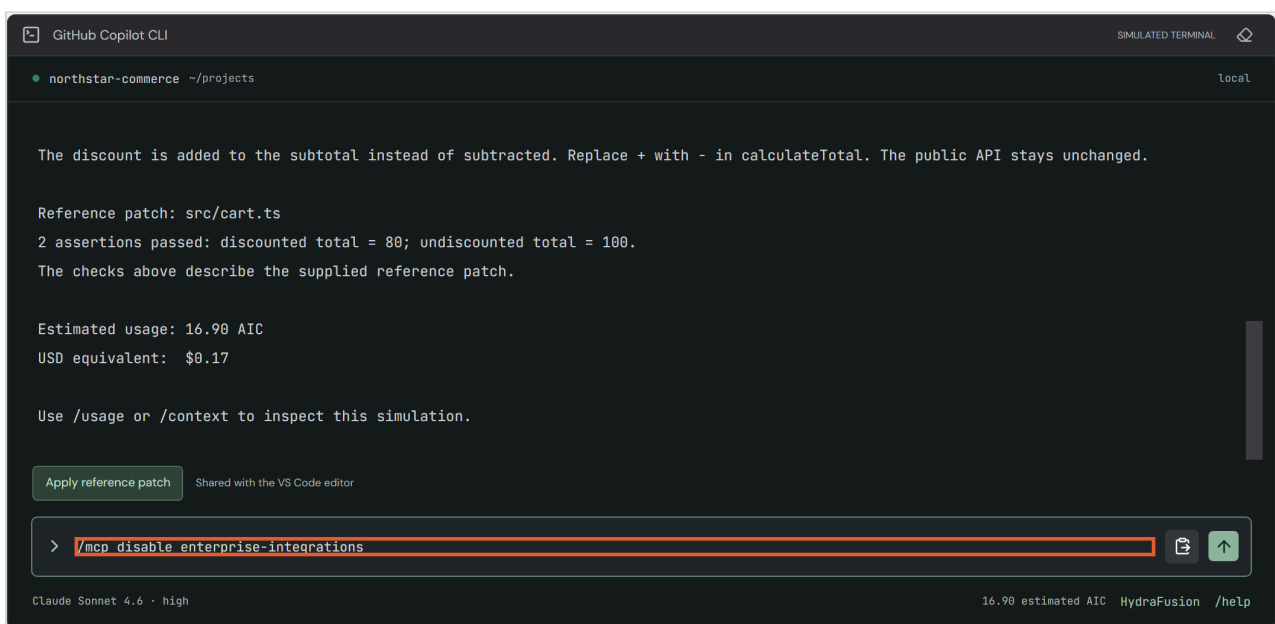
CHECK BEFORE CONTINUING

Trim enabled tools is on. Only the three core workspace tools remain; the fictional 25-tool integration group is disabled. **Guided value: 16.90 → 15.15 AIC.**

Manual CLI alternative

Type this command into the CLI input and press **Enter**. The terminal confirms that three core schemas remain.

```
/mcp disable enterprise-integrations
```



GitHub Copilot CLI

northstar-commerce ~/projects

local

The discount is added to the subtotal instead of subtracted. Replace `+` with `-` in `calculateTotal`. The public API stays unchanged.

Reference patch: `src/cart.ts`

2 assertions passed: discounted total = 80; undiscounted total = 100.

The checks above describe the supplied reference patch.

Estimated usage: 16.90 AIC

USD equivalent: \$0.17

Use `/usage` or `/context` to inspect this simulation.

Apply reference patch

Shared with the VS Code editor

> `/mcp disable enterprise-integrations`

Claude Sonnet 4.6 · high

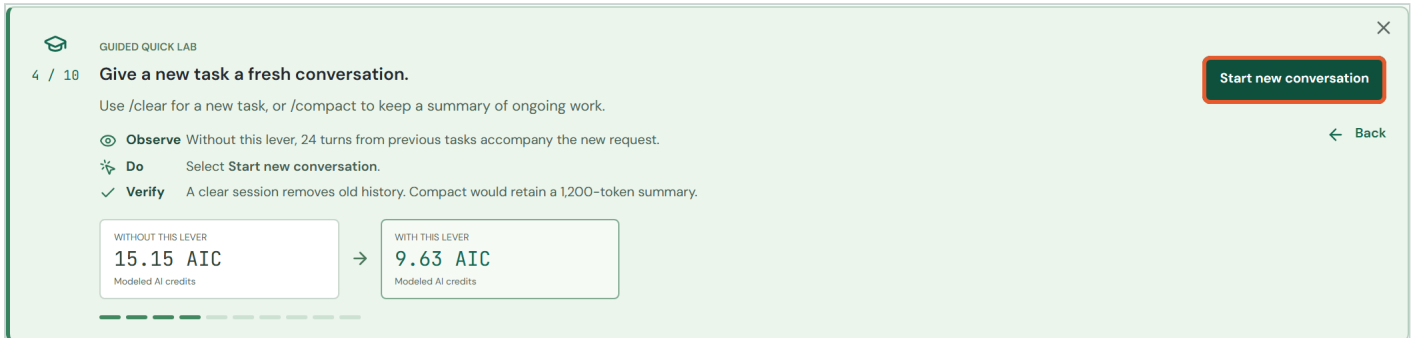
16.90 estimated AIC HydraFusion /help

Manual route: the highlighted input or picker is the control to use.

`enterprise-integrations` is fictional. The modeled saving assumes eager schema loading; deferred loading in real clients can make this saving smaller or zero.

Start a fresh conversation

1. In the green tour panel, click **Start new conversation**.
2. Wait for **Continue** to become enabled. Check the result, then click **Continue** once.



Guided CLI route: task 4 / 10.

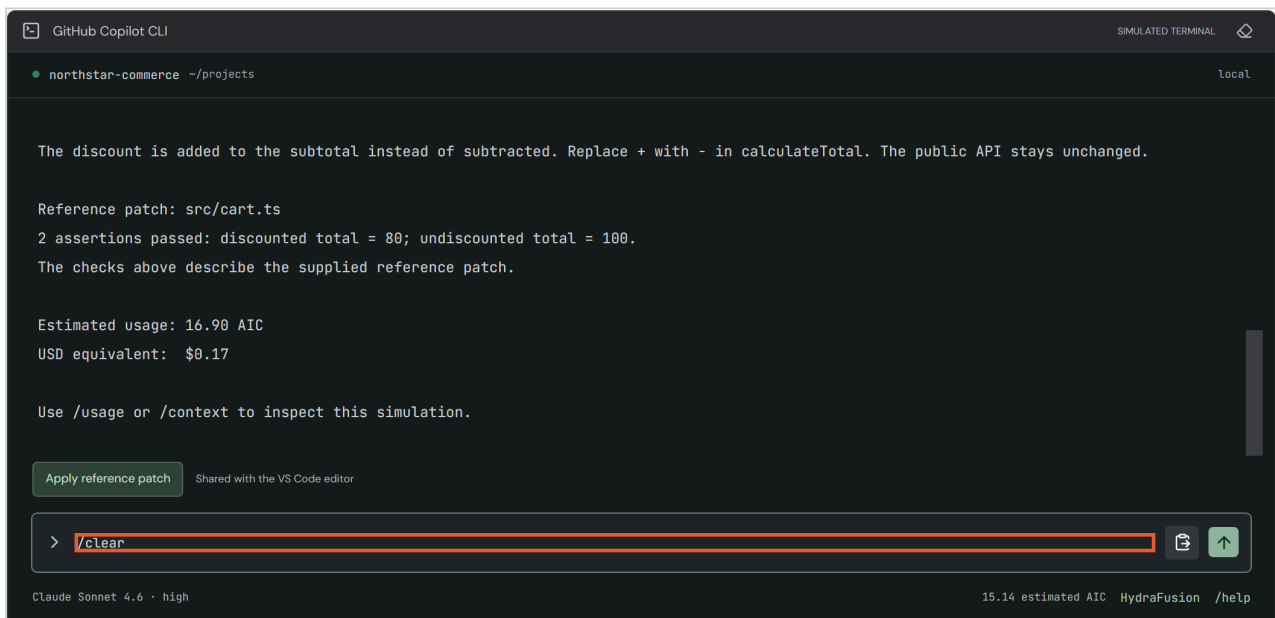
CHECK BEFORE CONTINUING

Refresh the session is on. The History category becomes **0.00 AIC**. The old response disappears, but the saved baseline run remains in Run history. **Guided value: 15.15 → 9.63 AIC**.

Manual CLI alternative

Type the following and press **Enter**. The terminal clears and its header reports **History: new session**.

```
/clear
```

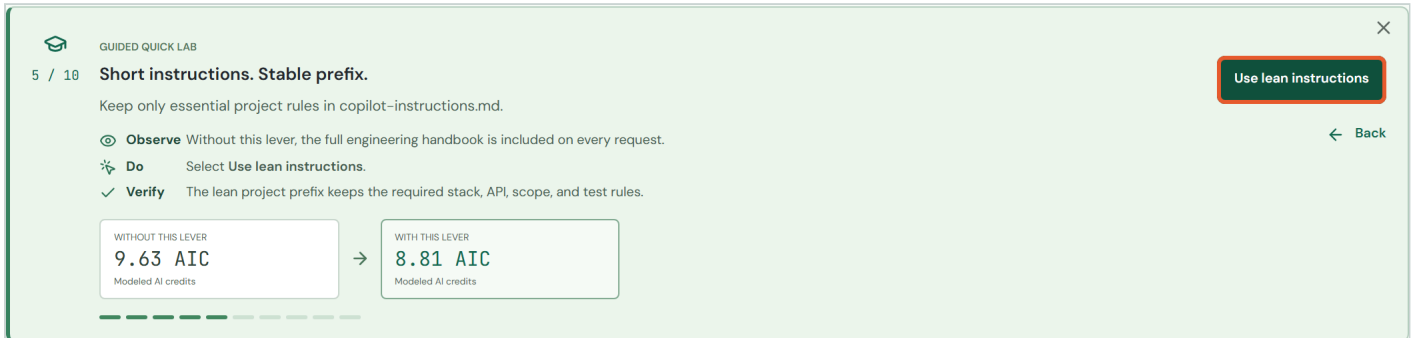


Manual route: the highlighted input or picker is the control to use.

For ongoing work, `/compact` retains a 1,200-token summary. It is not equivalent to `/clear` and produces a higher modeled total. The eraser icon only clears the displayed terminal entries.

Use lean project instructions

1. In the green tour panel, click **Use lean instructions**.
2. Wait for **Continue** to become enabled. Check the result, then click **Continue** once.



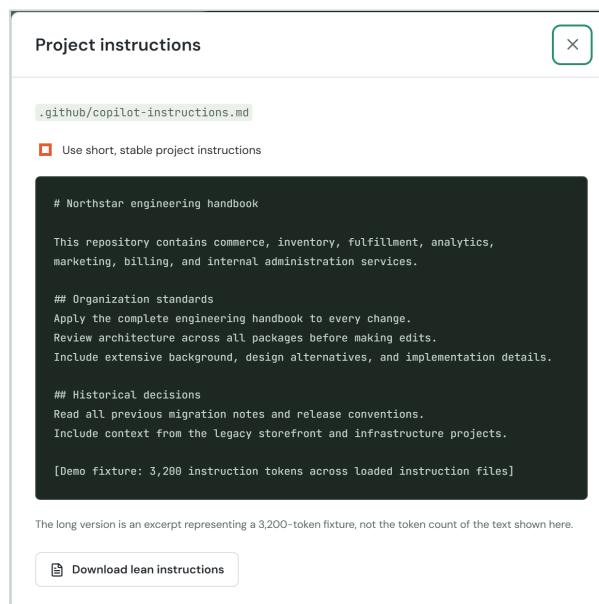
Guided CLI route: task 5 / 10.

CHECK BEFORE CONTINUING

Keep instructions lean is on. The fixture retains the stack, public-API constraint, focused reads, relevant tests, and minimal patch rules. **Guided value: 9.63 → 8.81 AIC.**

Manual CLI alternative

Enter **/instructions**. In the Project instructions dialog, **check Use short, stable project instructions**. Close it with the top-right X. Downloading the instructions alone does not enable this lever.



Manual route: the highlighted input or picker is the control to use.

Leave the warm-cache checkbox off for this walkthrough. Shorter instructions do not guarantee a provider cache hit.

Set medium reasoning effort

1. In the green tour panel, click **Set medium effort**.
2. Wait for **Continue** to become enabled. Check the result, then click **Continue** once.

GUIDED QUICK LAB

6 / 10 **Match reasoning effort to the task.**

This is a bounded regression with an explicit failing assertion, so medium effort is appropriate.

Observe Without this lever, the fixture spends a high reasoning budget exploring alternatives.

Do Select **Set medium effort**.

Verify Medium effort lowers modeled output while preserving the supplied reference fix.

WITHOUT THIS LEVER
8.81 AIC
Modeled AI credits

→

WITH THIS LEVER
4.01 AIC
Modeled AI credits

Set medium effort

Back

Guided CLI route: task 6 / 10.

CHECK BEFORE CONTINUING

Right-size reasoning is on. Output + reasoning decreases. The model may still be Claude Sonnet 4.6 until the next task. **Guided value: 8.81 → 4.01 AIC.**

Manual CLI alternative

Enter **/model**. In the picker above the CLI input, open **Effort**, select **Medium**, then click **Done**. Keep Model unchanged for this task.

Model: Claude Sonnet 4.6

Effort: High

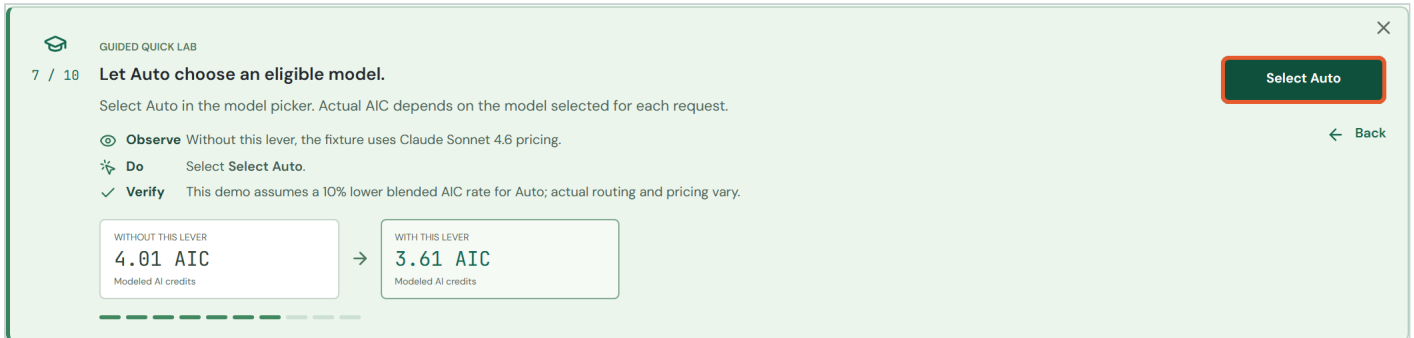
Done

Manual route: the highlighted input or picker is the control to use.

The response is a fixed reference answer. This does not prove that lower reasoning effort preserves quality on arbitrary real tasks.

Select Auto model routing

1. In the green tour panel, click **Select Auto**.
2. Wait for **Continue** to become enabled. Check the result, then click **Continue** once.



Guided CLI route: task 7 / 10.

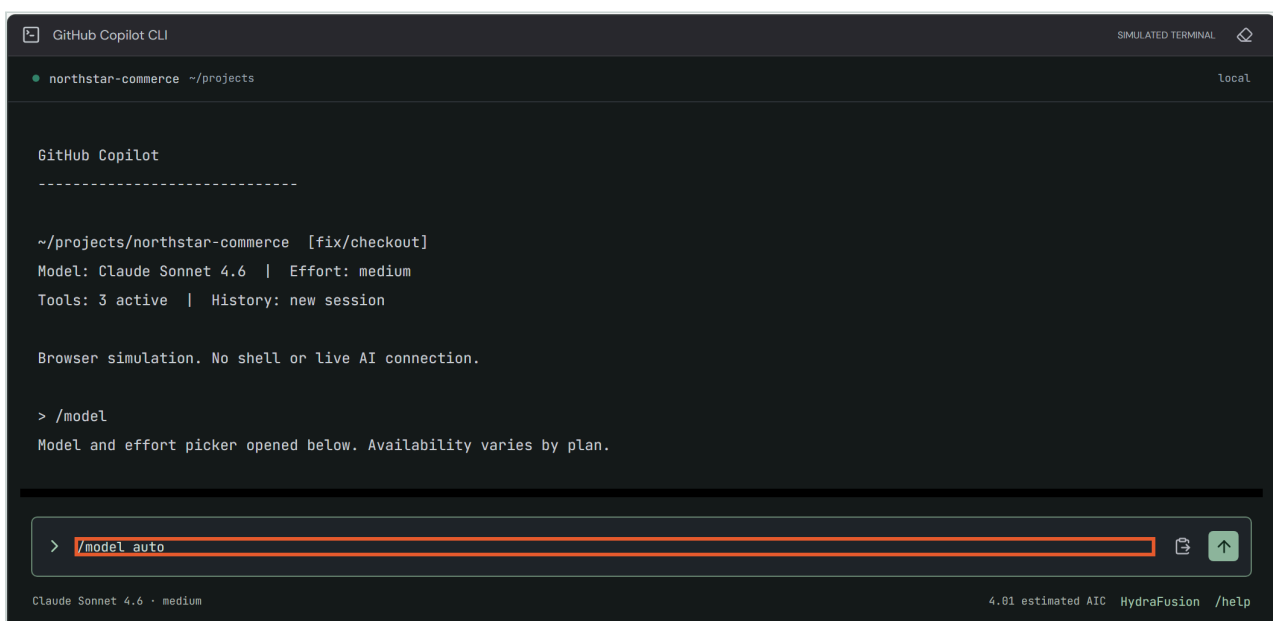
CHECK BEFORE CONTINUING

Let Auto route is on, all **6 / 6** levers are active, and the tour configuration reaches **3.61 AIC**.
Guided value: 4.01 → 3.61 AIC.

Manual CLI alternative

Enter the following command. The terminal confirms **Model set to auto**. The footer shows **Auto** and **medium**.

```
/model auto
```



Manual route: the highlighted input or picker is the control to use.

The simulator assumes a 10% lower blended rate for Auto. This is not a guaranteed discount or a statement about the model real Auto will select.

Run HydraFusion

1. At **Orchestrate the task with HydraFusion.**, click **Run HydraFusion** in the green panel.
2. Wait for the workflow result. Do not click again while it is running.
3. Verify the result below, then click **Continue**.

GUIDED QUICK LAB

8 / 10 **Orchestrate the task with HydraFusion.**

Run `/hydrafusion run` to simulate coordinator, explorer, implementer, and verifier stages sharing a compact task brief.

Observe The CLI shows each simulated agent stage and explicitly identifies the workflow as a fixture.

Do Select `Run HydraFusion`.

Verify A completed HydraFusion workflow saves the optimized CLI run and reports its modeled AIC.

WITHOUT LEVERS
23.80 AIC
Modeled AI credits

→

LAB GOAL
3.61 AIC
Modeled AI credits

[Run HydraFusion](#) [Back](#)

Task 8 / 10: the tour triggers the same fixture as `/hydrafusion run`.

CHECK BEFORE CONTINUING

The workflow lists coordinator, explorer, implementer, and verifier stages. The result is **3.61 AIC**; a fully optimized CLI run is saved.

Manual alternative

Enter `/hydrafusion run` in the input and press Enter. After completion, enter `/hydrafusion status`. The terminal should report **Latest HydraFusion workflow: completed**.

```

GitHub Copilot CLI
SIMULATED TERMINAL
northstar-commerce ~/projects local

The discount is added to the subtotal instead of subtracted. Replace + with - in calculateTotal. The public API stays unchanged.

Reference patch: src/cart.ts
2 assertions passed: discounted total = 80; undiscounted total = 100.
The checks above describe the supplied reference patch.

Estimated usage: 3.61 AIC
USD equivalent: $0.04

Use /usage or /context to inspect this simulation.

[Apply reference patch] Shared with the VS Code editor

> Enter a task or /help

Auto - medium 3.61 estimated AIC HydraFusion /help
  
```

Completed HydraFusion workflow and status in the deployed CLI.

What actually happened

HydraFusion enables all six levers and runs the supplied fixture. It is not a seventh lever and does not launch live agents, tools, or real tests. Apply the patch and run reference checks separately.

Recognize the saved optimized run

1. At **Same task. Now run it lean.**, look at the primary tour button.
2. After HydraFusion it already says **Continue**. Click Continue once. You do not need to find or press Run optimized task.
3. If you skipped HydraFusion and genuinely see **Run optimized task**, click it, wait for completion, then click Continue.

GUIDED QUICK LAB

9 / 10 **Same task. Now run it lean.**

Rerun the same scenario with all six levers. Compare the packet and reference patch, not just the headline percentage.

- Observe** The saved baseline shows how this task looked without optimization.
- Do** Select **Run optimized task**.
- Verify** The optimized run is saved beside the baseline with the same reference outcome.

WITHOUT LEVERS
23.80 AIC
Modeled AI credits

→

LAB GOAL
3.61 AIC
Modeled AI credits

Continue →

← **Back**

The verified post-HydraFusion state: Continue is already available.

CHECK BEFORE CONTINUING

A completed all-six-lever CLI run already exists. In the exact guided route there are two saved runs, not three. A separate extra run is optional and changes the count.

GitHub Copilot CLI

northstar-commerce ~/projects local

The discount is added to the subtotal instead of subtracted. Replace + with - in calculateTotal. The public API stays unchanged.

Reference patch: src/cart.ts

2 assertions passed: discounted total = 80; undiscounted total = 100.

The checks above describe the supplied reference patch.

Estimated usage: 3.61 AIC

USD equivalent: \$0.04

Use /usage or /context to inspect this simulation.

Apply reference patch Shared with the VS Code editor

> Enter a task or /help

Auto · medium 3.61 estimated AIC HydraFusion /help

Manual route example: /usage shows three runs because the file-scoped prompt also executed a request.

Do not confuse preview with a saved result

/usage shows the current configuration and completed-run count. Run history is the place to compare frozen results from completed requests.

Finish the CLI tour

- 1. Click **Finish tour** in the green panel.
- 2. Click **Run history** in the top page navigation. Confirm baseline and optimized rows.
- 3. Click **Simulator** to return. The next section applies and checks the reference patch.

 GUIDED QUICK LAB

10 / 10

Take the habit back to your workspace.
You have compared two modeled approaches to the same task. Export the run report, then validate gains with your own usage and quality checks.
Observe Review each category rather than treating the headline reduction as a guarantee.
Do Select Finish tour.
Verify Run history now contains a reproducible before-and-after comparison.

WITHOUT LEVERS
23.80 AIC
Modeled AI credits

→

LAB GOAL
3.61 AIC
Modeled AI credits

Finish tour

← Back

Task 10 / 10: Finish tour.

 Your demo runs 2

Export report

Run / scenario	Workspace	Levers	Baseline AIC	Configured AIC	AIC reduction
#2 Fix a checkout regression Cold cache · New-chat preset	 CLI	6 / 6	23.80	3.61	85%
#1 Fix a checkout regression Cold cache · New-chat preset	 CLI	0 / 6	23.80	23.80	0%

Each row compares the same scenario and prompt with all levers off versus the captured configuration. AIC estimates are frozen when a run completes and are not billing telemetry.

Verified CLI guided results: 0 / 6 baseline and 6 / 6 optimized rows.

CHECK BEFORE CONTINUING

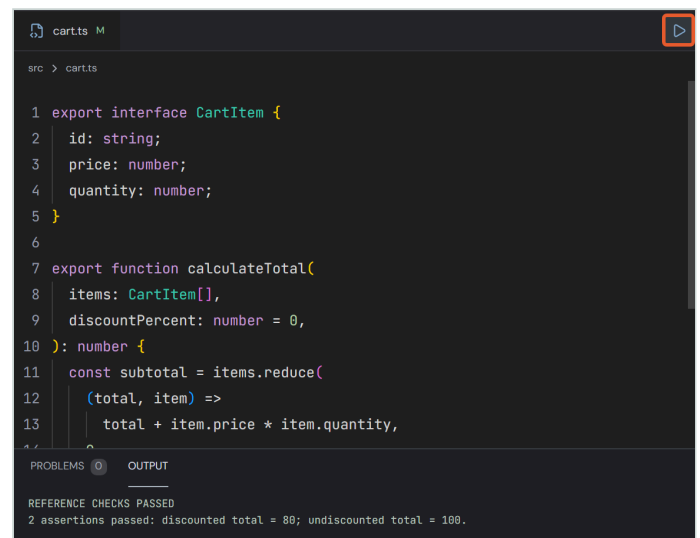
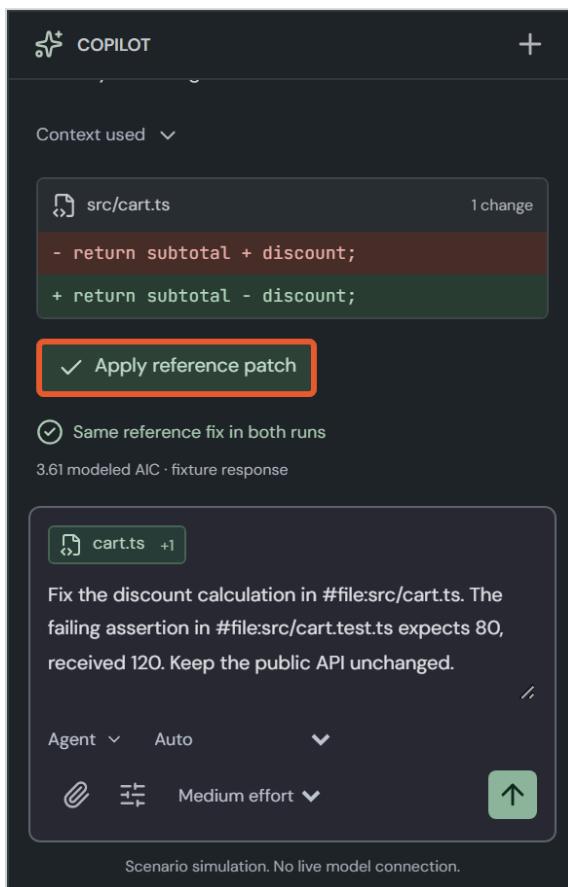
The optimized row shows **3.61 Configured AIC** and **85% AIC reduction**. The new-chat, cold-cache configuration remains active.

A small baseline difference is expected

The original run displays 23.80 AIC. The optimized row's Baseline AIC can display 23.81 because it recalculates an unoptimized baseline using that row's longer focused prompt. The guide does not round these into a claim of measured savings.

Apply the patch and check it

1. From Simulator, click **VS Code** above the workspace. A response from the CLI is shared with the editor.
2. In Copilot Chat, click **Apply reference patch**. If it says Reference patch applied, skip this click.
3. Click the small triangle at the right edge of the editor tab bar, accessible name **Run reference checks**, tooltip **Check reference fixture**. This is not the Send prompt arrow.
4. Read the **OUTPUT** panel underneath the editor.



2. Run reference checks in the editor tab bar; read OUTPUT below.

1. Apply reference patch in Copilot Chat.

CHECK BEFORE CONTINUING

REFERENCE CHECKS PASSED

2 assertions passed: discounted total = 80; undiscounted total = 100.

Reference checks, not arbitrary execution

The simulator recognizes its supplied source fixture. It does not run a real Vitest process or execute custom edits. If you see CUSTOM EDIT DETECTED, reapply the reference patch before checking again.

Read the saved results correctly

- 1. Click **Run history** in the page header.
- 2. Read **Levers**, **Baseline AIC**, **Configured AIC**, and **AIC reduction** for each row.
- 3. The newest run is at the top. The exact guided route gives two rows; manual prompts or repeated runs add more.
- 4. Click **Simulator** to return. Click **Comparison** in the inspector for the current unsaved configuration.

Your demo runs 2

Export report

Run / scenario	Workspace	Levers	Baseline AIC	Configured AIC	AIC reduction
#2 Fix a checkout regression Cold cache - New-chat preset	CLI	6 / 6	23.80	3.61	85%
#1 Fix a checkout regression Cold cache - New-chat preset	CLI	0 / 6	23.80	23.80	0%

Each row compares the same scenario and prompt with all levers off versus the captured configuration. AIC estimates are frozen when a run completes and are not billing telemetry.

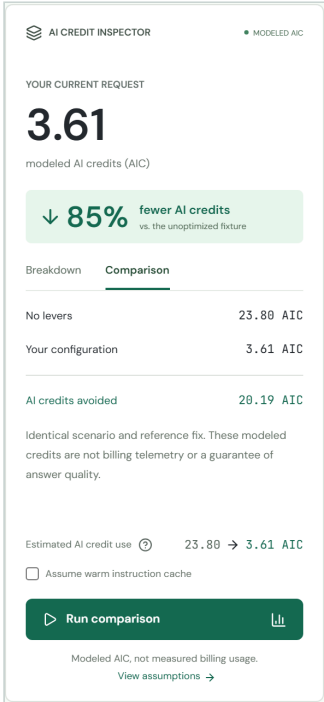
Saved run table: results are frozen when each request completes.

Saved versus current

Changing switches updates the inspector immediately but does not change old rows. Run comparison or Send prompt to save a new row.

For the checkout route, the headline is approximately **85% fewer modeled AIC**, with the same supplied reference answer.

The optimized row compares its own prompt against all levers off. It is not a direct subtraction from every other row.



Current configuration comparison, not historical billing.

Quality is a separate check. Equal answers are built into this fixture. For real work, measure actual usage and independently validate output quality.

Download the run report

- 1. Complete at least one run. Open **Run history**.
- 2. Click **Export report** at the upper-right of the results table.
- 3. Open your browser Downloads list. Look for **copilot-ai-credit-lab-report.json**. It is JSON data, not this PDF.
- 4. You can also hover over the download-arrow icon near the page heading and click **Export run report**. Both controls export the same report.

🕒 Your demo runs 2

⬇️ Export report

Run / scenario	Workspace	Levers	Baseline AIC	Configured AIC	AIC reduction
#2 Fix a checkout regression Cold cache - New-chat preset	📁 CLI	6 / 6	23.80	3.61	85%
#1 Fix a checkout regression Cold cache - New-chat preset	📁 CLI	0 / 6	23.80	23.80	0%

Each row compares the same scenario and prompt with all levers off versus the captured configuration. AIC estimates are frozen when a run completes and are not billing telemetry.

Primary export control: Export report in the table heading.

GITHUB COPILOT > HANDS-ON LAB

AI credit optimization lab.

Same task. Same reference fix. A more intentional request.

No credentials needed

📄 ⚙️ ⬇️ ↶ ↷

Alternative export control: highlighted download icon beside the page heading.

CHECK BEFORE CONTINUING

The live browser download was verified as **copilot-ai-credit-lab-report.json**, containing two runs from the guided CLI route.

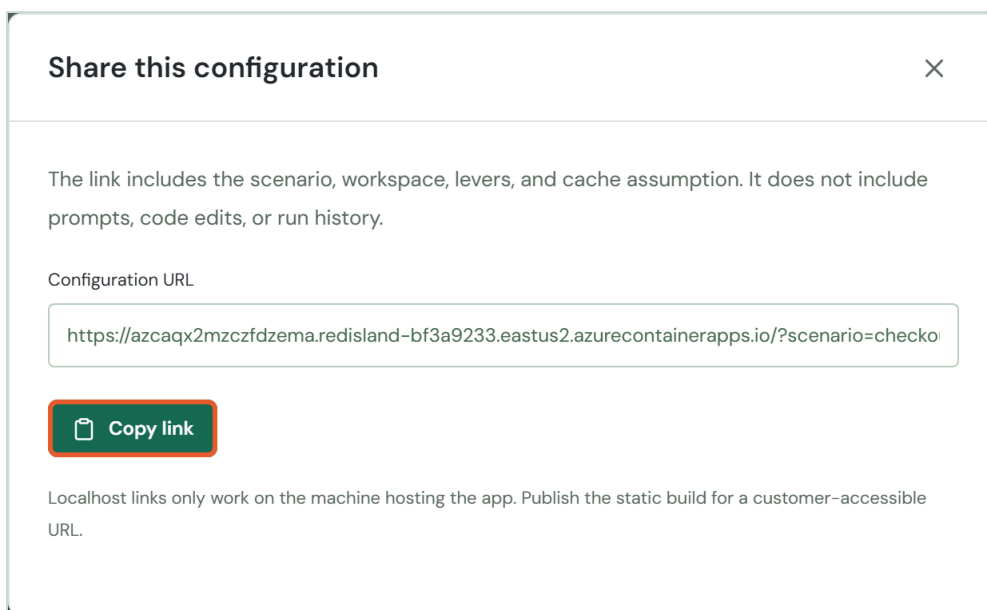
Review before sharing

Exports include entered prompts, settings, scenario results, and assumptions. Use synthetic examples and review the JSON before sharing. A configuration link does not substitute for an exported run report.

Disabled button? No completed run exists yet. Run the baseline and wait for its simulated response, then retry the export. If the browser blocks a download, allow the download for this site.

Send a reproducible configuration

1. Click **Simulator** and leave the desired scenario, surface, and lever settings selected.
2. In the upper-right page action icons, click the connected-nodes icon, tooltip **Share configuration**.
3. In **Share this configuration**, click **Copy link**. Expect **Link copied**.
4. If clipboard access is blocked, click the Configuration URL field, press **Ctrl+C** on Windows, and paste it into Teams.
5. Ask the recipient to open the link and run the request. Attach your exported JSON separately when they need your saved run evidence.



Share dialog: Copy link copies the current configuration URL.

CHECK BEFORE CONTINUING

The URL includes scenario, surface, lever choices, and applicable cache/session settings. It does not include code edits, custom prompts, or run history.

Starting the tour resets shared settings

A recipient can inspect the restored configuration immediately. Clicking Guided tour intentionally replaces it with a new unoptimized baseline. Share the report as well when preserving the results matters.

Fix an API retry boundary

1. Export previous runs first. Exit any open guided tour with its X.
2. Select **Fix an API retry boundary** under **YOUR SCENARIO**, then click **VS Code**. Click **Task brief** to inspect the problem and close it with X.
3. For the full before/after exercise, click **Guided tour** and repeat the nine VS Code tasks. Scenario selection alone does not clear previous levers or history.
4. For a quick fix-only exercise instead: turn all six switches on (or click **Enable all** if shown), keep warm cache off, and click **Send prompt**.
5. When the response appears, click **Apply reference patch**, then the editor-tab **Run reference checks** triangle.

Your task brief

BACKEND · TYPESCRIPT / VITEST

Fix an API retry boundary

The inventory client makes four requests when the configured maximum is three. Fix the retry boundary while preserving the error behavior.

Repository	northstar-inventory
Implementation	src/retry.ts
Regression test	src/retry.test.ts

All files are demo fixtures. Both approaches have the same reference fix. Keep the task constant while changing the request context.

Task brief: implementation and regression-test files.

retry.ts M

src > retry.ts

```

1 export async function retry<T>({
2   operation: () => Promise<T>,
3   maxAttempts: number = 3,
4 }): Promise<T> {
5   let lastError: unknown;
6
7   for (let attempt = 0;
8     attempt < maxAttempts;
9     attempt++) {
10    try {
11      return await operation();
12    } catch (error) {
13      lastError = error;

```

PROBLEMS 0 OUTPUT

REFERENCE CHECKS PASSED

2 assertions passed: exactly 3 calls; original final error preserved.

Reference patch applied; OUTPUT shows the expected assertions.

CHECK BEFORE CONTINUING

REFERENCE CHECKS PASSED

2 assertions passed: exactly 3 calls; original final error preserved.

With all six levers, clear history, default focused prompt, and cold cache: **3.72 AIC**.

Why the fix works: The inclusive loop runs attempts 0, 1, 2, and 3. The reference change uses < instead of <= to make exactly three attempts while preserving the final error.

CLI variant: select this scenario, switch to Copilot CLI, and enter /hydrafusion run. It automatically uses src/retry.ts and src/retry.test.ts.

Cover a shipping edge case

1. Export previous runs first. Exit any open guided tour with its X.
2. Select **Cover a shipping edge case** under **YOUR SCENARIO**, then click **VS Code**. Click **Task brief** to inspect the problem and close it with X.
3. For the full before/after exercise, click **Guided tour** and repeat the nine VS Code tasks. Scenario selection alone does not clear previous levers or history.
4. For a quick fix-only exercise instead: turn all six switches on (or click **Enable all** if shown), keep warm cache off, and click **Send prompt**.
5. When the response appears, click **Apply reference patch**, then the editor-tab **Run reference checks** triangle.

Your task brief

TEST COVERAGE · TYPESCRIPT / VITEST

Cover a shipping edge case

Orders of exactly \$50 should qualify for free shipping, but they do not. Pinpoint the boundary and add regression coverage.

Repository	northstar-fulfillment
Implementation	src/shipping.ts
Regression test	src/shipping.test.ts

All files are demo fixtures. Both approaches have the same reference fix. Keep the task constant while changing the request context.

Task brief: implementation and regression-test files.

shipping.ts

```

1 export function shippingCost(
2   orderTotal: number,
3 ): number {
4   const freeShippingThreshold = 50;
5   const standardShipping = 5.99;
6
7   if (orderTotal >= freeShippingThreshold) {
8     return 0;
9   }
10
11   return standardShipping;
12 }
13

```

PROBLEMS 0

OUTPUT

REFERENCE CHECKS PASSED

3 assertions passed: 49.99 → 5.99; 50 → 0; 50.01 → 0.

Reference patch applied; OUTPUT shows the expected assertions.

CHECK BEFORE CONTINUING

REFERENCE CHECKS PASSED

3 assertions passed: 49.99 → 5.99; 50 → 0; 50.01 → 0.

With all six levers, clear history, default focused prompt, and cold cache: **3.50 AIC**.

Why the fix works: The reference change uses `>=` instead of `>` so exactly \$50 qualifies for free shipping. The boundary tests are already supplied in this demo; you do not need to author them.

CLI variant: select this scenario, switch to Copilot CLI, and enter `/hydrafusion run`. It automatically uses `src/shipping.ts` and `src/shipping.test.ts`.

What the simulator counts

Read Breakdown in the inspector

System: fixed system–instruction fixture.

Instructions: long or lean project rules.

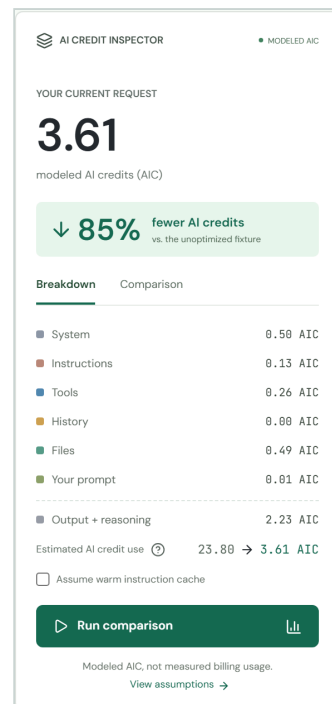
Tools: available tool schemas, not the number of tools actually executed.

History: old turns, an empty conversation, or a retained summary.

Files: broad retrieval or focused task files.

Your prompt: an estimate from entered character count.

Output + reasoning: fixed answer plus illustrative reasoning budget.



The current all–six–lever, cold–cache checkout request.

Pricing assumptions

The model converts estimated input and output usage to USD using its configured Sonnet 4.6 rates, then uses **1 AIC = \$0.01 USD**. Auto applies an illustrative 10% lower blended rate. These are simulator assumptions, not current–plan billing advice.

Optional cache experiment

1. Enable **Keep instructions lean** first. Otherwise the cache checkbox is disabled.
2. Check **Assume warm instruction cache** below the inspector. Observe the changed estimate, then uncheck it to restore the guide's values.
3. Click **View assumptions** below Run comparison for the complete methodology.

Not counted as measured operations

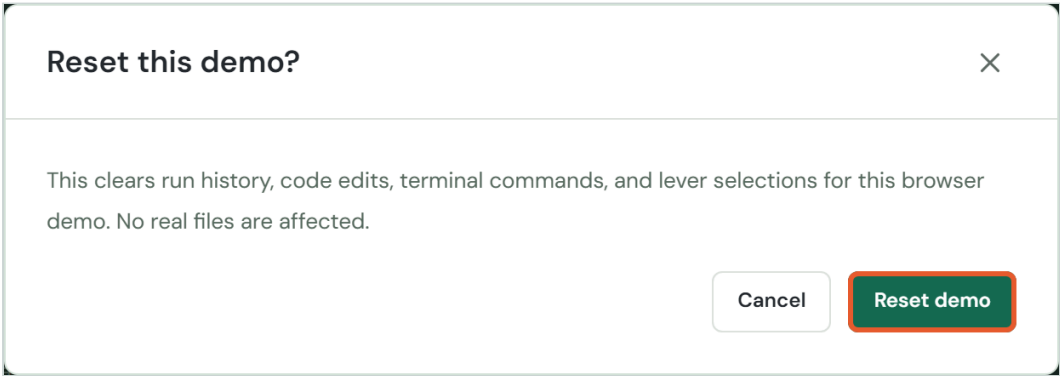
No live model request, actual tool execution, real agent workflow, billing telemetry, or real provider cache hit is measured. Compaction overhead is excluded. Hosting charges are not part of the AIC estimate.

Get unstuck without restarting blindly

What you see	What to do
Button already says Continue	The task is complete. Click Continue once. This is expected after HydraFusion.
Command input disabled	Wait for the simulation to finish. Do not repeatedly submit. Stop simulation in the inspector cancels the current run; cancelled runs are not saved.
AIC differs from the guide	Check the checkout scenario, default prompt, cold cache, clear history, and lever order. Manual prompts or /compact legitimately change totals.
No Apply reference patch button	Run a task and wait for its response. New chat clears the visible response; run the optimized task again if necessary.
REFERENCE CHECK FAILED	The original source is still present. Click Apply reference patch, then Run reference checks.
CUSTOM EDIT DETECTED	Custom code is not executed here. Reapply the reference patch to restore the recognized fixture.
Scenario selector is disabled	Exit the guided panel with its X and wait for any running request to complete.
Cannot see the editor on a narrow screen	In VS Code view, use the Editor / Copilot Chat pane tabs. Scroll down to reach the inspector and six switches.

Only when you need a complete restart

- 1. Export existing runs first.
- 2. Click the circular-arrow page icon, tooltip **Reset demo**.
- 3. Read the dialog and click **Reset demo** again to confirm, or **Cancel** to keep your work.



Reset confirmation: clears local runs, edits, terminal entries, and lever selections.

Reloading also loses in-memory runs. The URL can restore configuration, not results. Real Copilot versions, plans, available models, tools, and organization policy may differ from this replica.