

```
Project: End of Mod X SE 103_C1_Semeste X User Containers - w X Mandela Quote and X Google Gemini X Device Activation X bawumia23/set-se- X [GitHub Copilot Documentation]

Based on the documentation, here's how to authenticate with GitHub Copilot CLI:

On first launch:
- Run copilot and use the /login slash command to authenticate via the browser

Using a Personal Access Token (PAT):
1. Create a fine-grained PAT at https://github.com/settings/personal-access-tokens/new (https://github.com/settings/personal-access-tokens/new)
2. Enable "Copilot Requests" permission
3. Set GH_TOKEN or GITHUB_TOKEN environment variable with your token

Check current user:
- Use /user show to see who's logged in
- Use /user list and /user switch to manage multiple GitHub accounts

Is there a specific authentication task you'd like help with?

> "I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message."
• Check recent commits
$ cd /root/set-se-103 && git log --oneline -5
l 4 lines...

• I see. Actually, looking at your log, the last two commits (077fd56 and 6f70d9c) are already pushed to origin (they show origin/main). I need commits that don't have remote tracking.

Are you working on a different branch, or would you like to:
1. Combine commits on a new/unpushed branch (reset to an earlier point and squash)
2. Rewrite already-pushed history (requires force push, only if you have exclusive access)
3. Check if there are unpushed commits on a different branch

Which applies to your situation?

~/set-se-103/advanced-shell-practice[✗ main]
> Type @ to mention files or / for commands
shift+tab cycle mode
```

Accuracy Ranking: 10/10

Justification: The AI correctly identified that an interactive rebase (`git rebase -i HEAD~2`) is the standard industry practice for "squashing" local commits. This method allows the user to combine history while keeping the codebase clean before pushing to a shared repository.