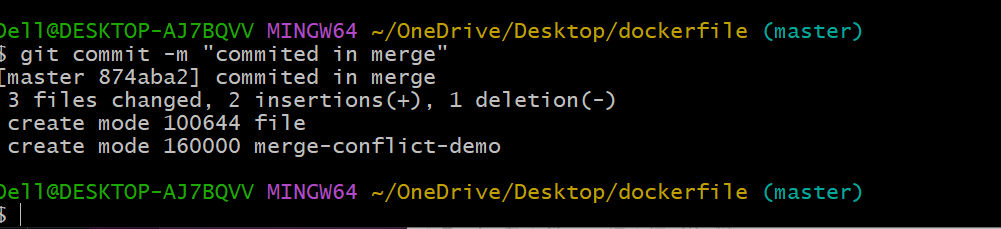
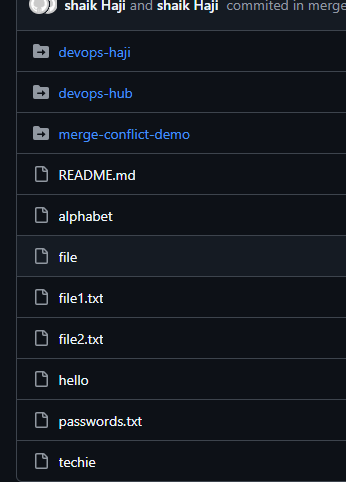
Git Challenges

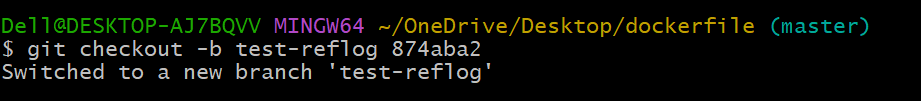
1. **Resolve Merge Conflicts**
   * Create a merge conflict intentionally (two users editing the same line).



* + Resolve the conflict and push the changes.



1. **Recover Deleted Branch**
   * Delete a local branch and then recover it using the reflog.



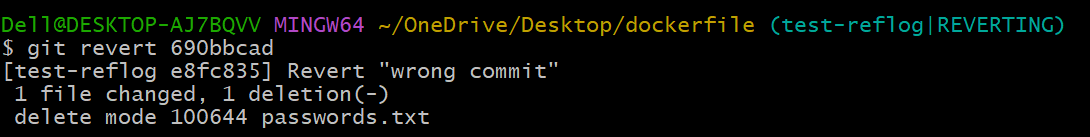
· Delete branch → git branch -d test-reflog

· Inspect history → git reflog

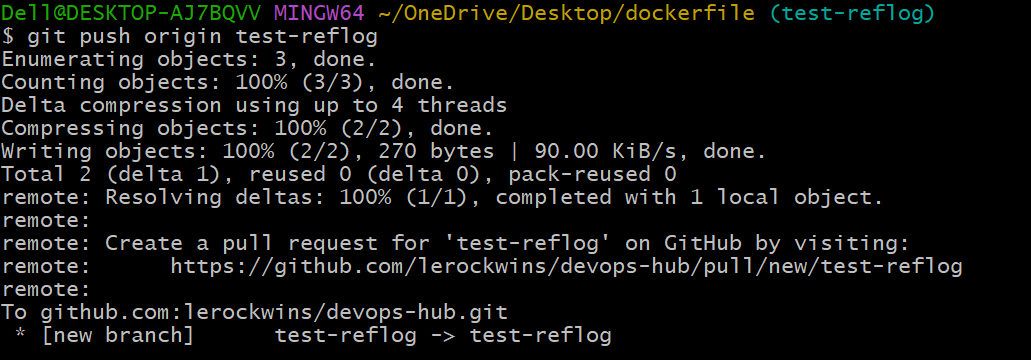
· Recreate branch → git checkout -b test-reflog <commit-hash>

1. **Undo Wrong Push**
   * Push a wrong commit to GitHub, then undo it without losing history.

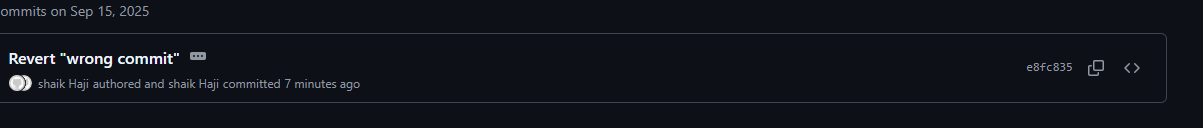
* Using a command of git revert and hash id
* And open a interface and chnge the wrong commit



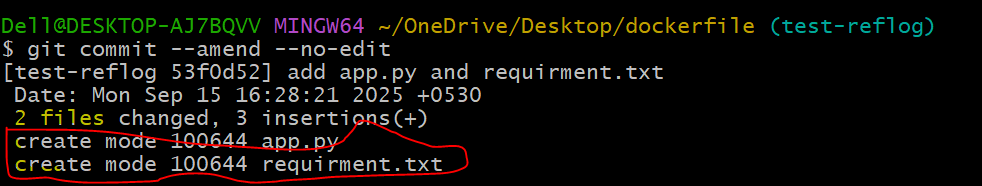
* Then git push origin and branch name
* Then the pull request is acceptable and it taken in to github.



* Then gave a pull request and then undo it.



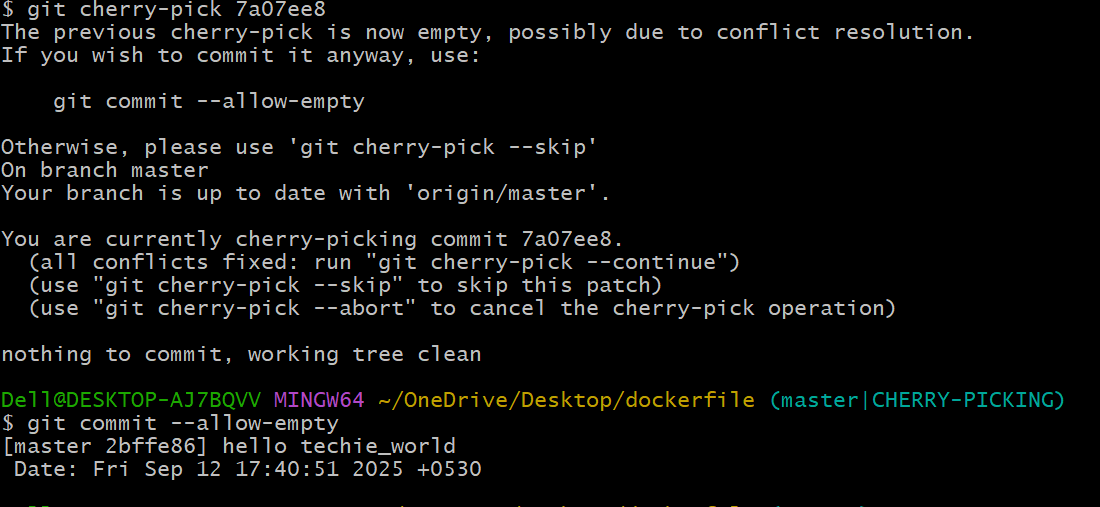
1. **Amend a Commit**
   * Make a commit, then add a missing file to it using git commit --amend.



* Create a 2 files echo and different commit id.
* Then add the 2 files git add .
* Then check the status if it’s a add or not.
* And after git commit --amend
* And finally git commit --amend --no-edit then the two files in one commit id and message will be same.

1. **Cherry-pick a Commit**
   * Take a specific commit from one branch and apply it to another branch.

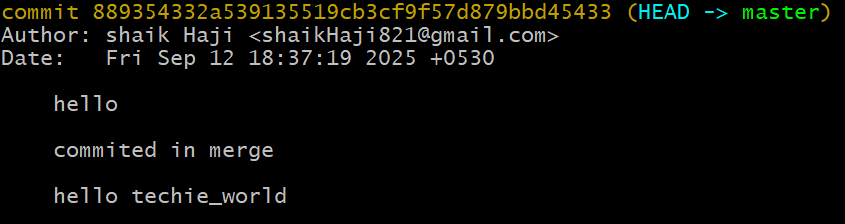
* If uh want the cherry-pick any commit use a command for git cherry-pick and which want chane gave a commit id numers.
* And it is showing me some hints to I used a command of “git commit --allow-empty”
* Then after gave push to central repo and pull request ND here the result of cherry-pick result.



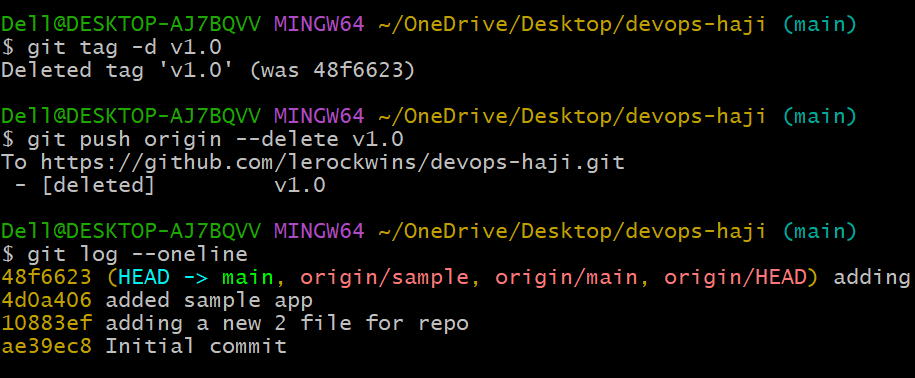


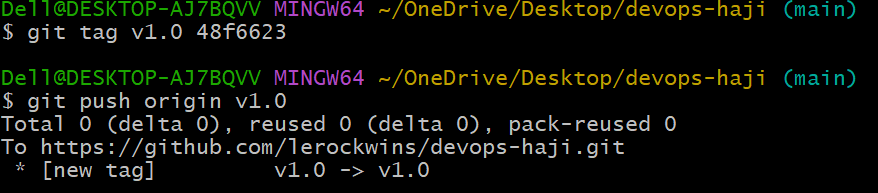
1. **Interactive Rebase**
   * Reorder and squash multiple commits into a single clean commit.

* to sqaush a commit file
* To use a command of git log --oneline
* Then use a command of git rebase-i head~3
* And it will pick a top 3 files of log and it will open a new interface where we chnge the commit messege 3 in to 1
* So we will leave a first one pick command
* Then other to pick replace with squash.

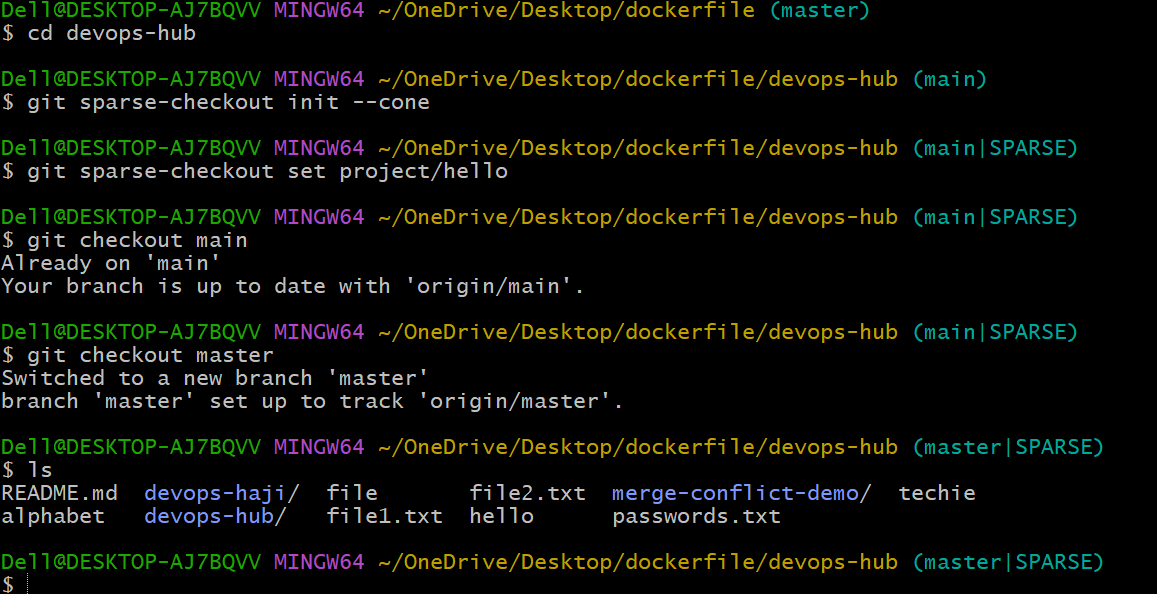


1. **Tagging & Release**
   * Create a version tag (v1.0), push it to GitHub, then delete and restore it.

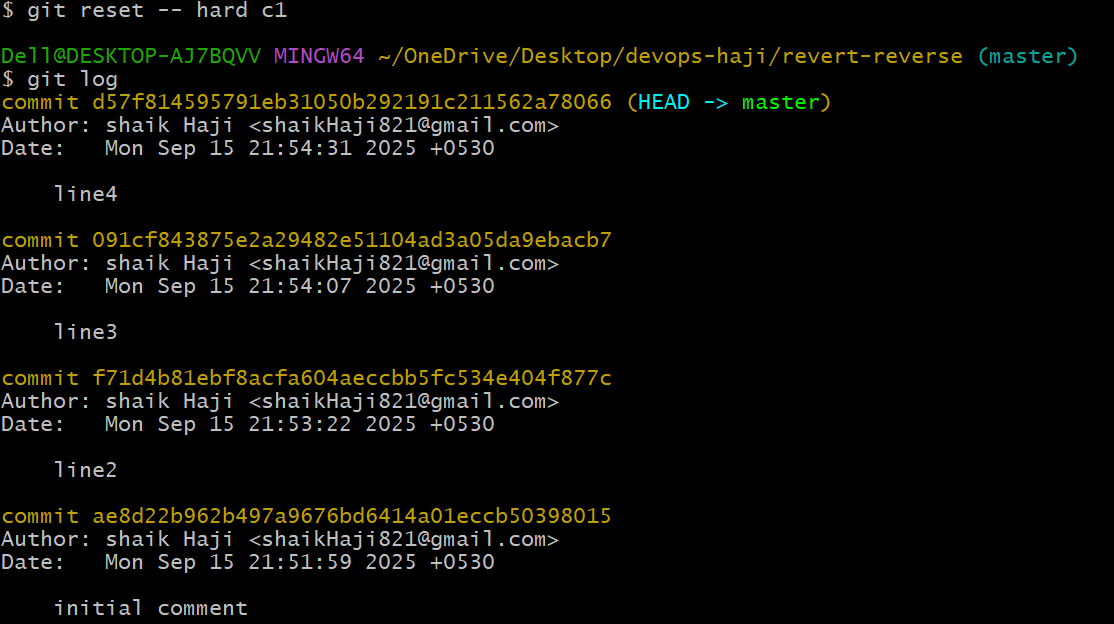


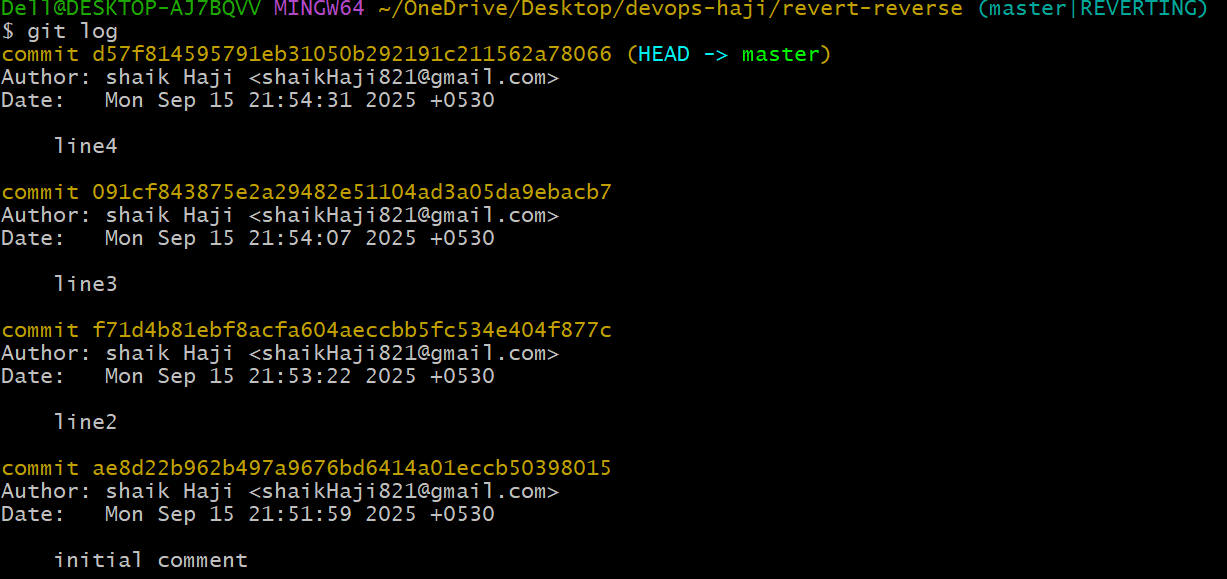


1. **Clone with Sparse Checkout**
   * Clone only a subdirectory of a repo using sparse checkout.

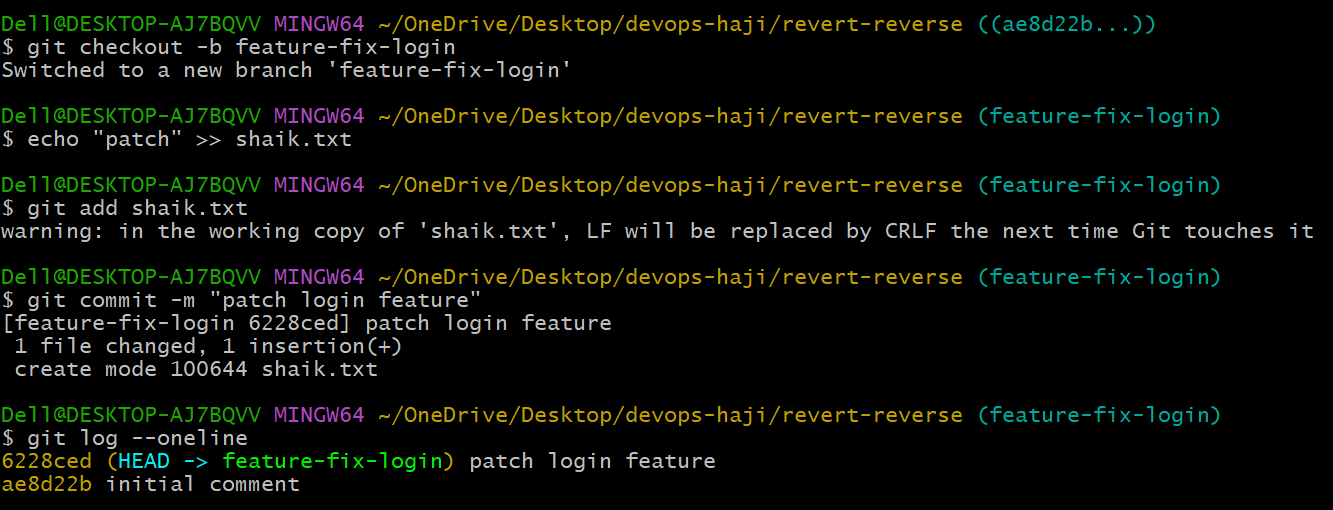


1. **Reset vs Revert Challenge**
   * Demonstrate the difference between git reset --hard and git revert in a repo.

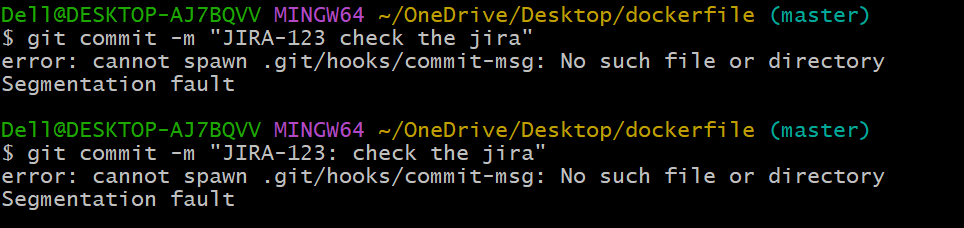




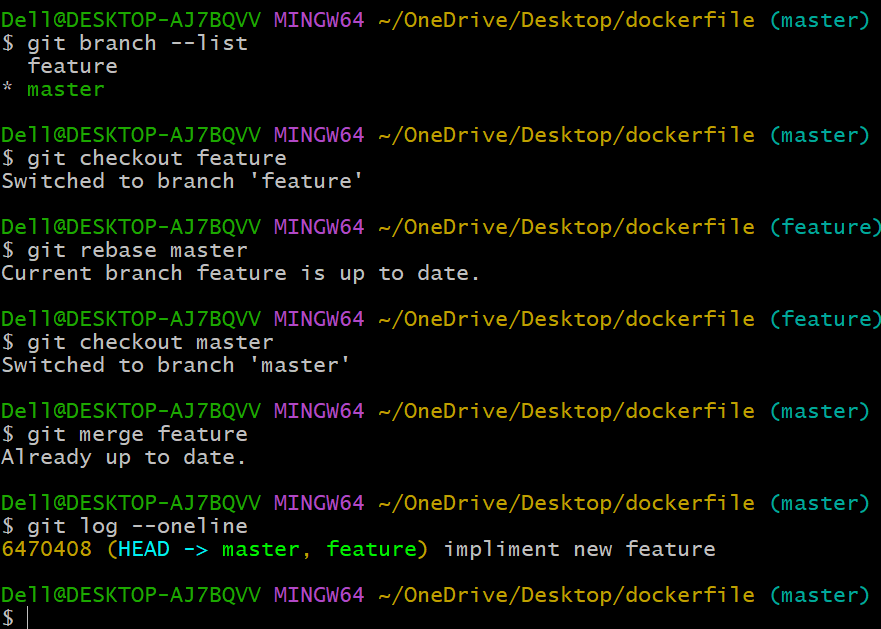
1. **Detached HEAD Challenge**
   * Checkout a specific commit (detached HEAD state) and create a new branch from it.

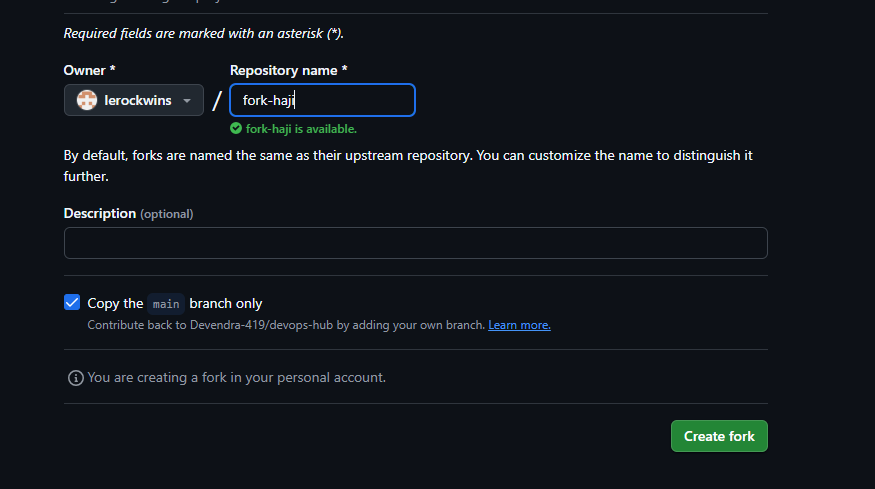


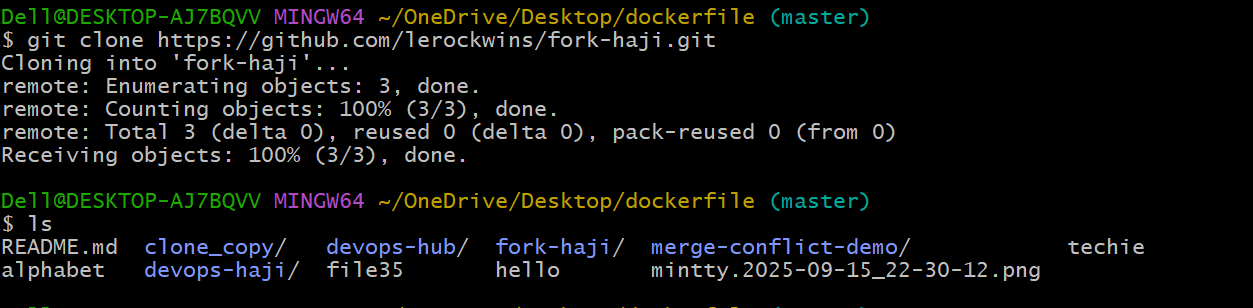
1. **Git Hooks Challenge**
   * Configure a pre-commit hook to reject commits without a message format (e.g., must start with JIRA-XXX).

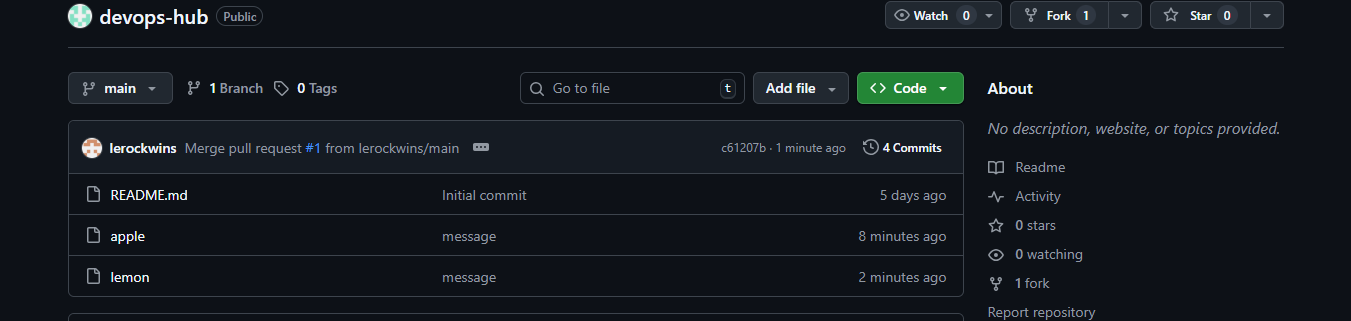


1. **Squash Merge vs Rebase Merge**
   * Show the difference between squash merge and rebase merge with evidence.



1. **Fork & Pull Request Workflow**
   * Fork a repo, make a change, and submit a pull request to the original repo.
2. 





1. **Recover Lost Commit**
   * Commit something, reset hard, and then recover it using git reflog.

