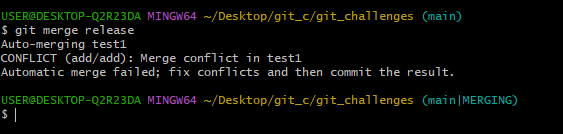
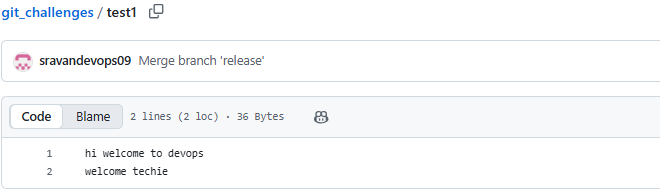
**GIT & GITHUB CHALLENGES**

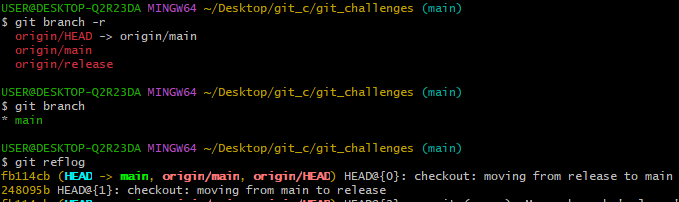
**1.Resolve Merge Conflicts**

**Create a merge conflict intentionally (two users editing the same line)**





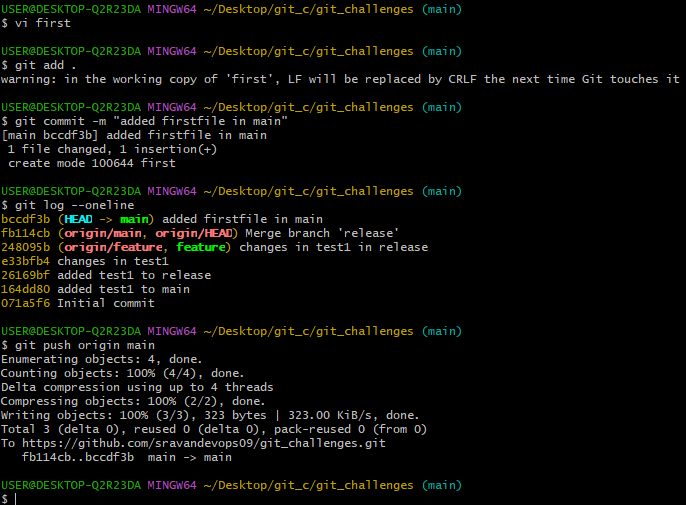
**Recover Deleted Branch**

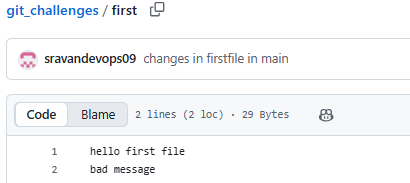


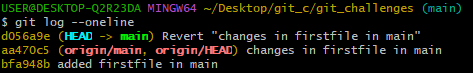


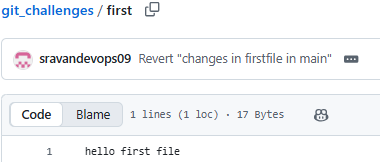
**Undo Wrong Push**

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









**Amend a Commit**

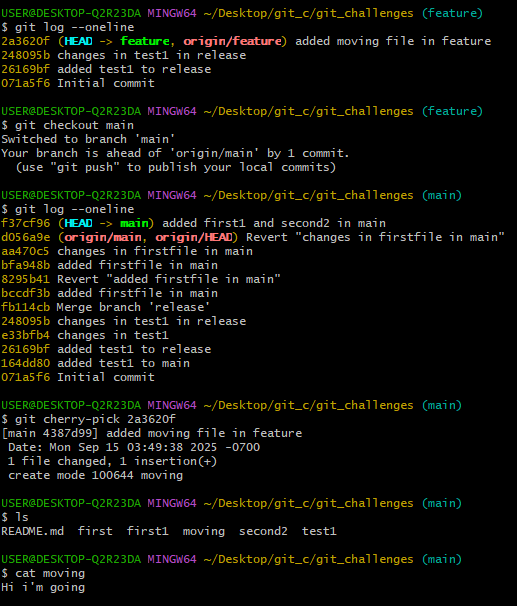
**Make a commit, then add a missing file to it using git commit --amend.**

**git commit –amend**



**Cherry-pick a Commit**

**Take a specific commit from one branch and apply it to another branch.**



**Interactive Rebase**

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

**git rebase -i HEAD~3**

**Tagging & Release**

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

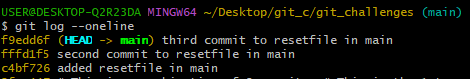
**Clone with Sparse Checkout**

**Clone only a subdirectory of a repo using sparse checkout.**

**Reset vs Revert Challenge**

**Demonstrate the difference between git reset --hard and git revert in a repo.**

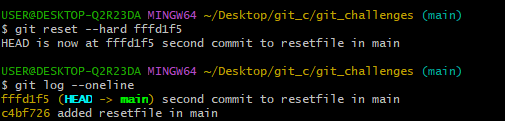
**Created 3 commits:**

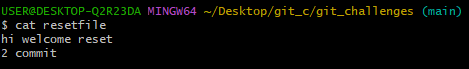


**git reset:**



Suppose we want to “undo” commit 3.

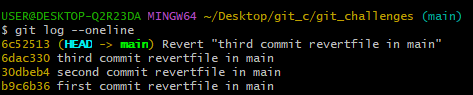




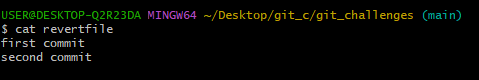
**git revert:**



**git revert 6dac330**

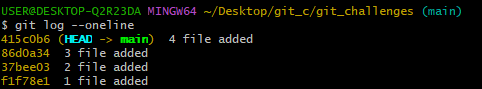


Commit 3 is still in history, but its effect is undone

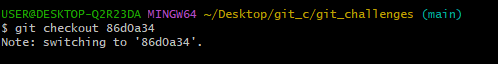


**Detached HEAD Challenge**

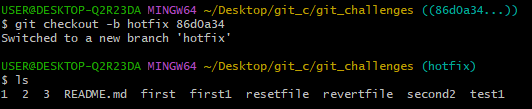
**Checkout a specific commit (detached HEAD state) and create a new branch from it.**



Pick an older commit hash:



It starts from commit :



**Squash Merge vs Rebase Merge**

**Show the difference between squash merge and rebase merge with evidence.**

