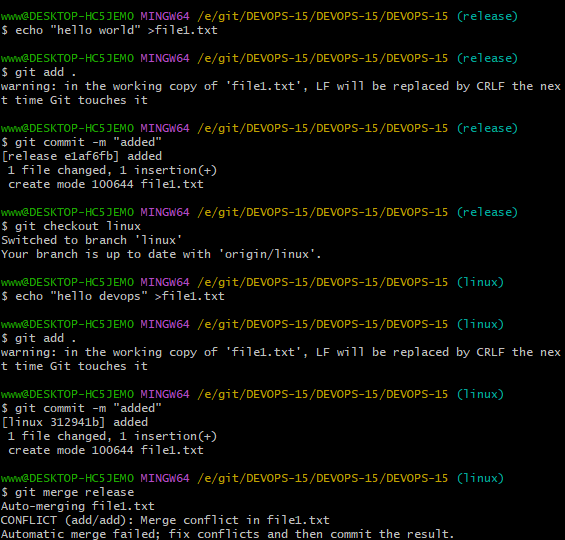
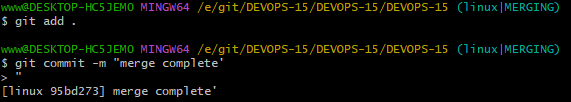
GIT CHALLENGE

1. Resolve merge conflicts.

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

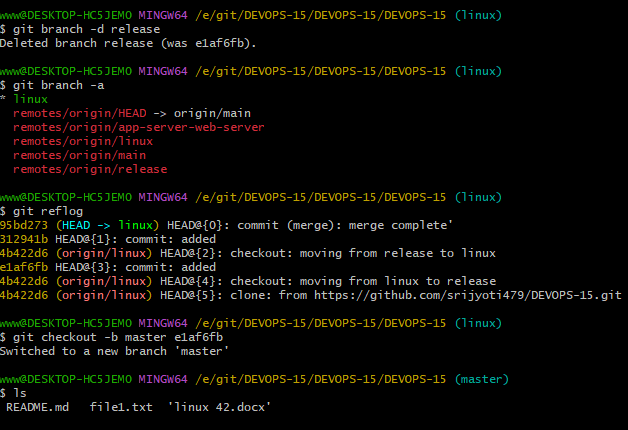
Resolve the conflict and push the changes.



2. Recover deleted branch.

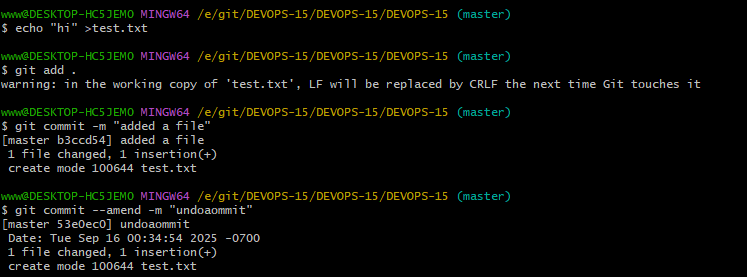
Delete a local branch and then recover it using the reflog

* Git branch -d <branch name >:to delete the branch
* Git log :check the log
* Gitb checkout -b master<commitid>: the deleted branch it will recover the deleted branch

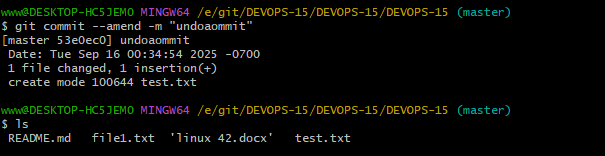


3.Undo wrong push

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

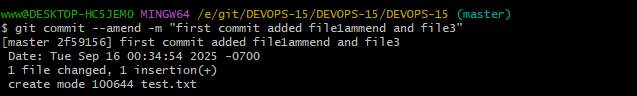


Use to find the commit hash you want to undo



4.Amend a commit

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



5.Cherry-pick a commit

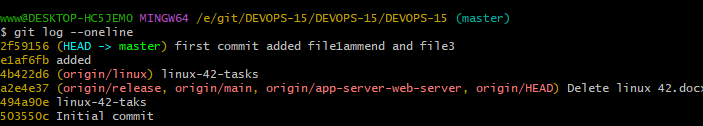
Take a specific commit from one branch and apply it to another branch

Git log-oneline (to check the commit id and copy commit id)

Switch to other branch and enter

Git cherry-pick (commit\_id)

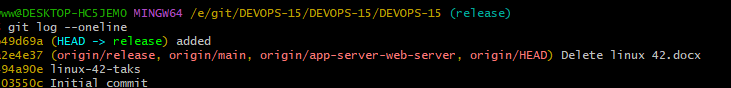
Git log -oneline (hence you can get the spicifc id in other branch)





6. Interactive rebase

Reorder and squash multiple commits into a single clean commit





7. Tagging &Release

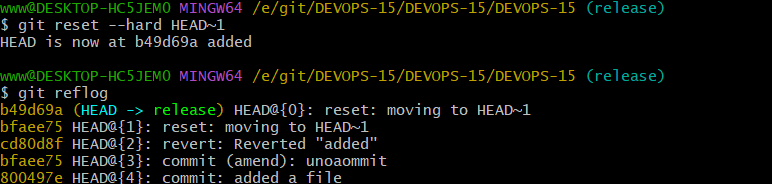
Create a version tag (vl.0), push it to github, then delete and restore it.

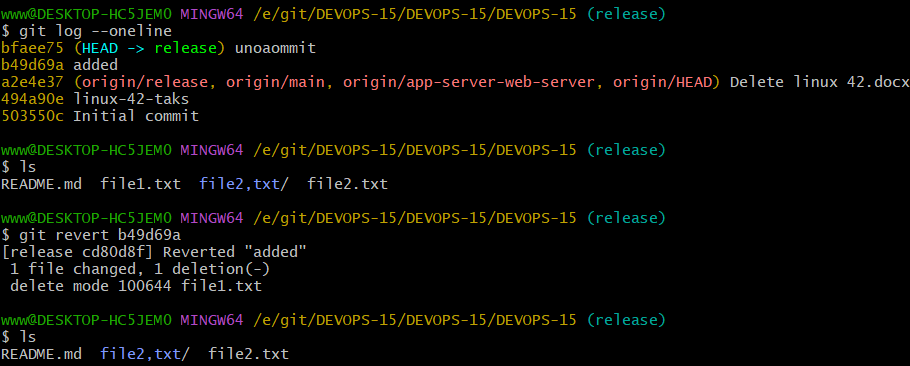
8. Clone with sparse checkout

Clone only a subdirectory of a repo using sparse checkout.

9. Reset vs Revert challenge

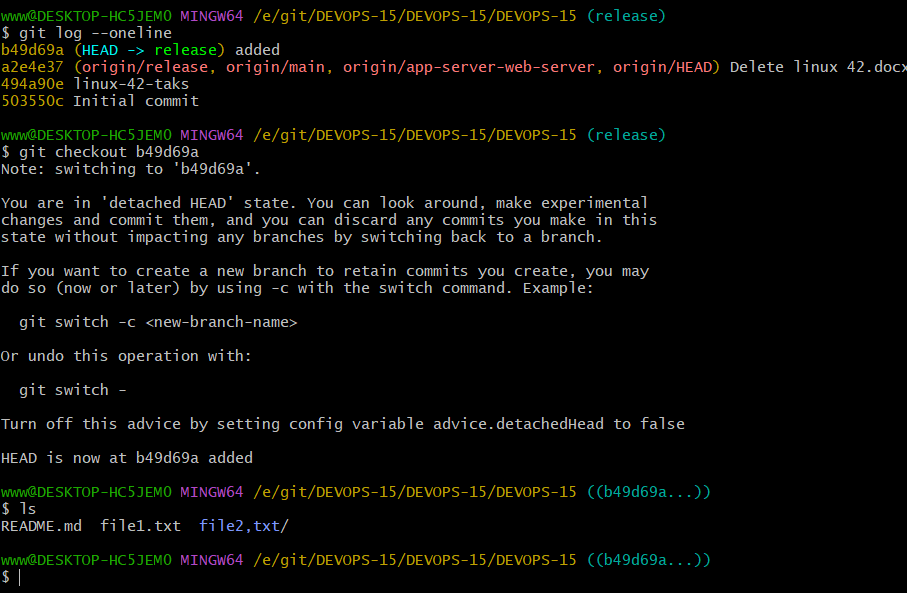
Demonstrate the difference between git reset—hard and git revert in a repo.





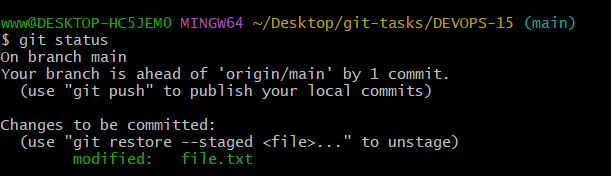
10. Detached HEAD challenge

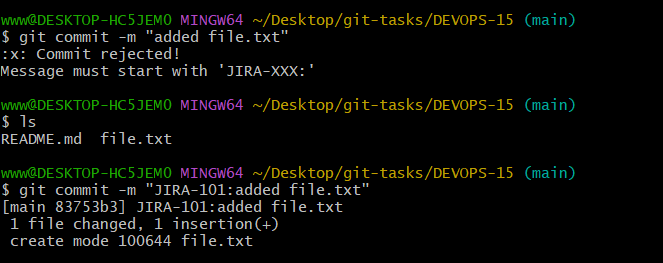
Checkout a specific commit (detached HEAD state) and create a new branch from it.

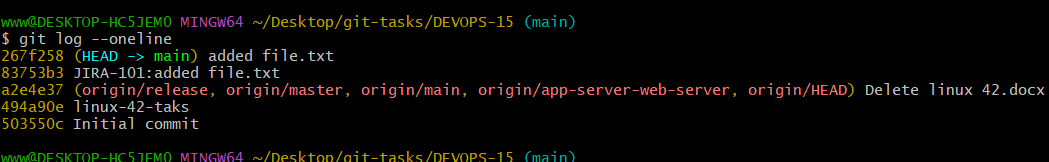


11.Git hooks challenge

Configure a pre-commit hook to reject commits without a message format (e.g., must start with JIRA-XXX).+





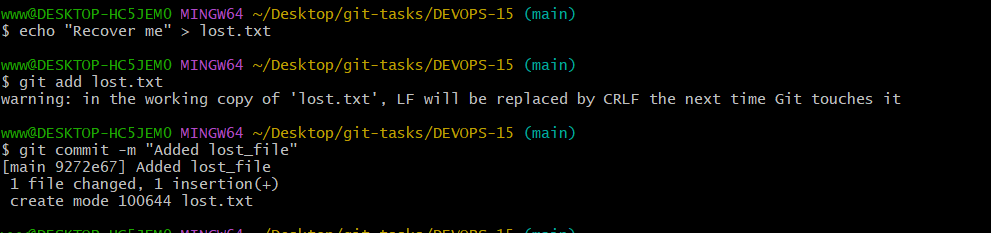


i .git/hooks/commit-msg  
paste these code  
#!/bin/bash  
# Comm it message file passed by Git  
commit\_msg\_file=$1

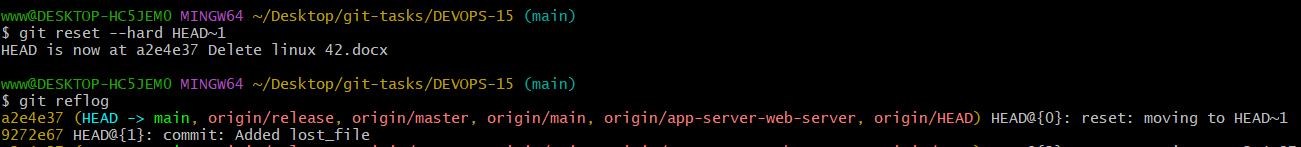
# Read first line of commit message  
commit\_msg=$(head -n1 "$commit\_msg\_file")  
# Regex: Commit must start with JIRA-followed by numbers, e.g., JIRA-123  
if ! [[ $commit\_msg =~ ^JIRA-[0-9]+: ]]; then  
 echo "❌Commit rejected!"  
 echo "Message must start with 'JIRA-XXX:'"  
 exit 1  
fi  
exit 0  
-------------------------------------------------------------------------------------------------------------------------------  
⦁ chmod 777 .git/hooks/commit-msg  
⦁ then created and add one file the commit on it   
⦁ it will show commit rejected (commit should start with JIRA-XXX

⦁ commit again-------git commit -m "JIRA-101:created file in main"  
⦁ hence it will create a commit sucessfully   
14. Recover lost commit.

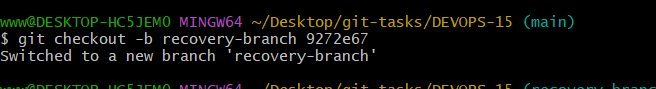
Create a file and commit it



Use git reset –hard HEAD~1 to revert one commit back and git reflog to list commit



Use git checkout -b recovery-branch commit\_id to create new branch at that commit



Use git reset-hard commit\_id to move head back to that commit

