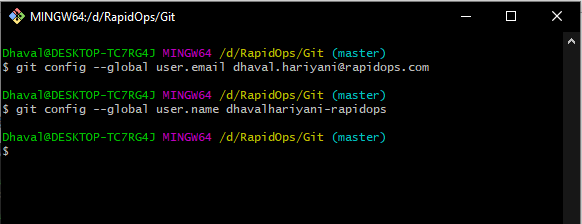
1. **Configure your** **user name and email.**

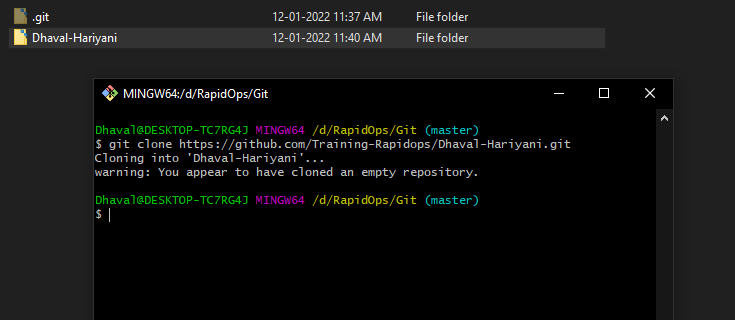
git config –global user.email dhaval.hariyani@rapidops.com

git config –global user.name dhavalhariyani-rapidops



1. **Clone repo of your name from GitHub to the local system.**

git clone https://github.com/Training-Rapidops/Dhaval-Hariyani.git



1. **Create a file inside the repo, and make your first commit "My First Commit".**

ls

cd Dhaval-Hariyani

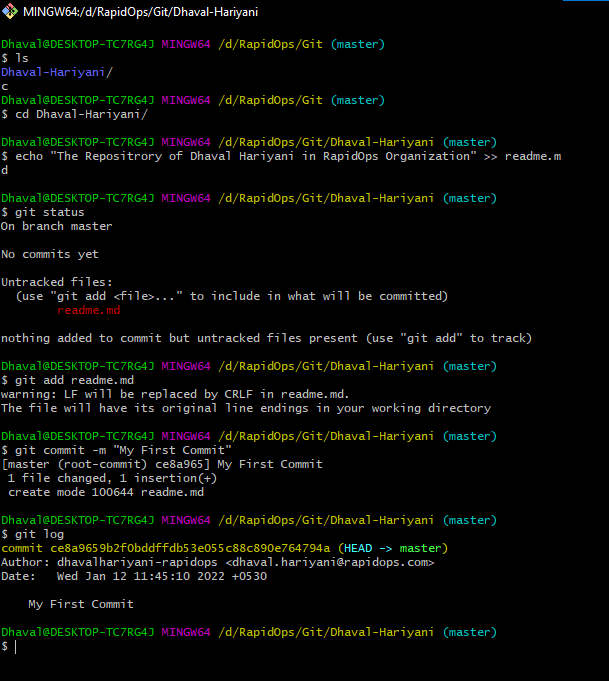
echo “The Repositrory of Dhaval Hariyani in RapidOps Organization” >> readme.md

git status

git add readme.md

git commit –m “My First Commit”

git log



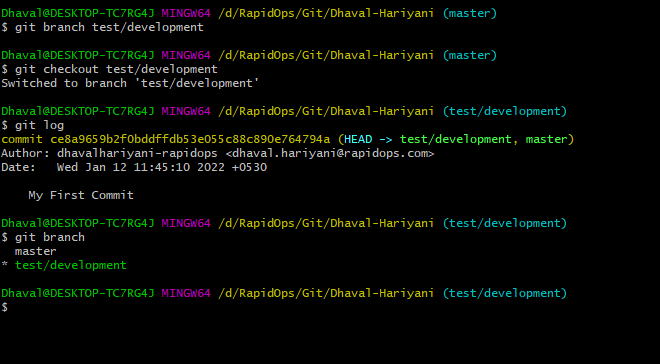
1. **Create and switch to the branch 'test/development' (create from the master branch and it should be from origin).**

git branch test/development

git checkout test/development

git log

git branch

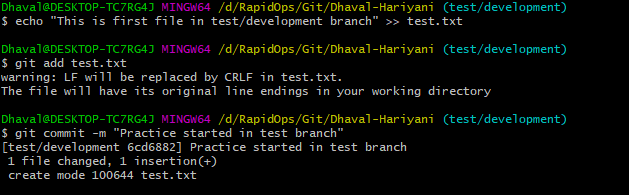


1. **Add a file in this branch and commit your changes with the message "Practice started in test branch".**

echo "This is first file in test/development branch" >> test.txt

git add test.txt

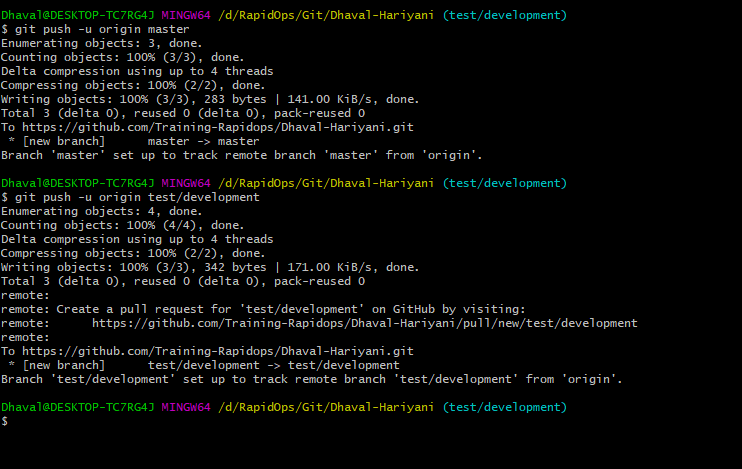
git commit -m "Practice started in test branch"

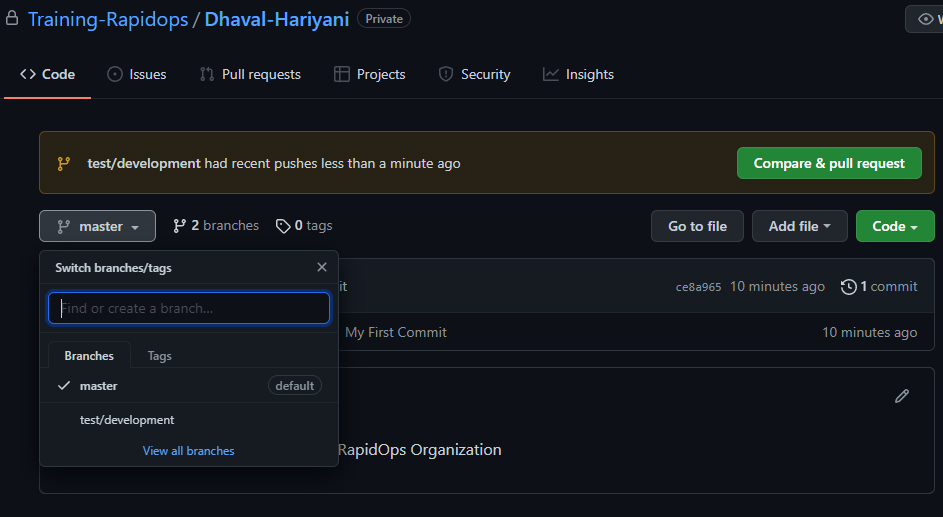


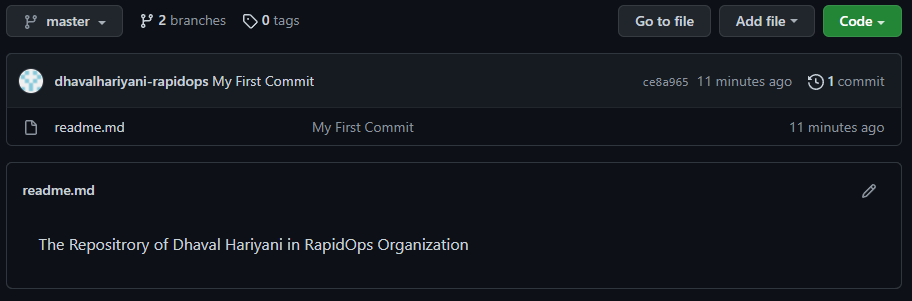
1. **Now push your changes and this branch to the remote.**

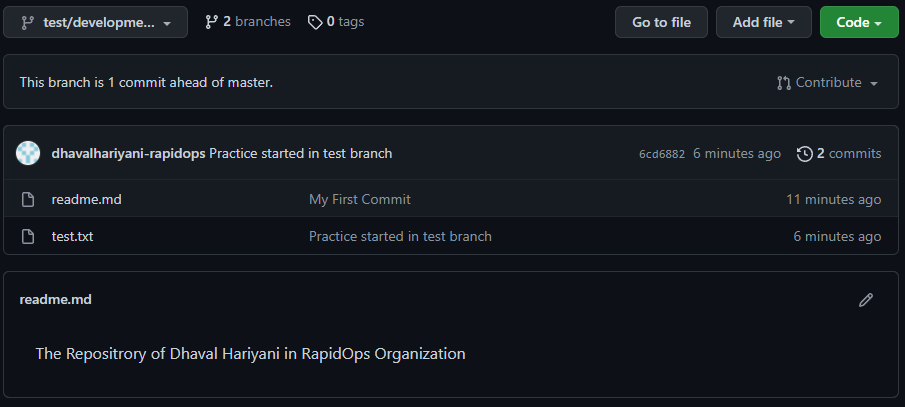
git push -u origin master

git push -u origin test/development

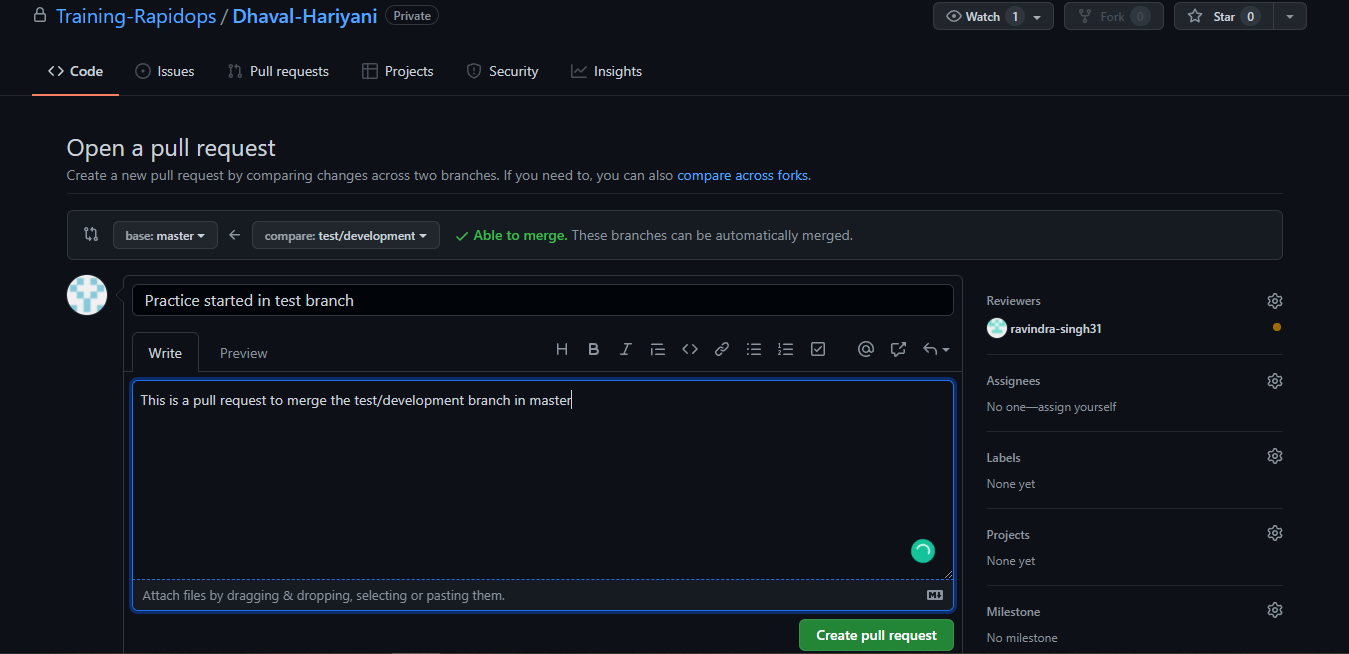


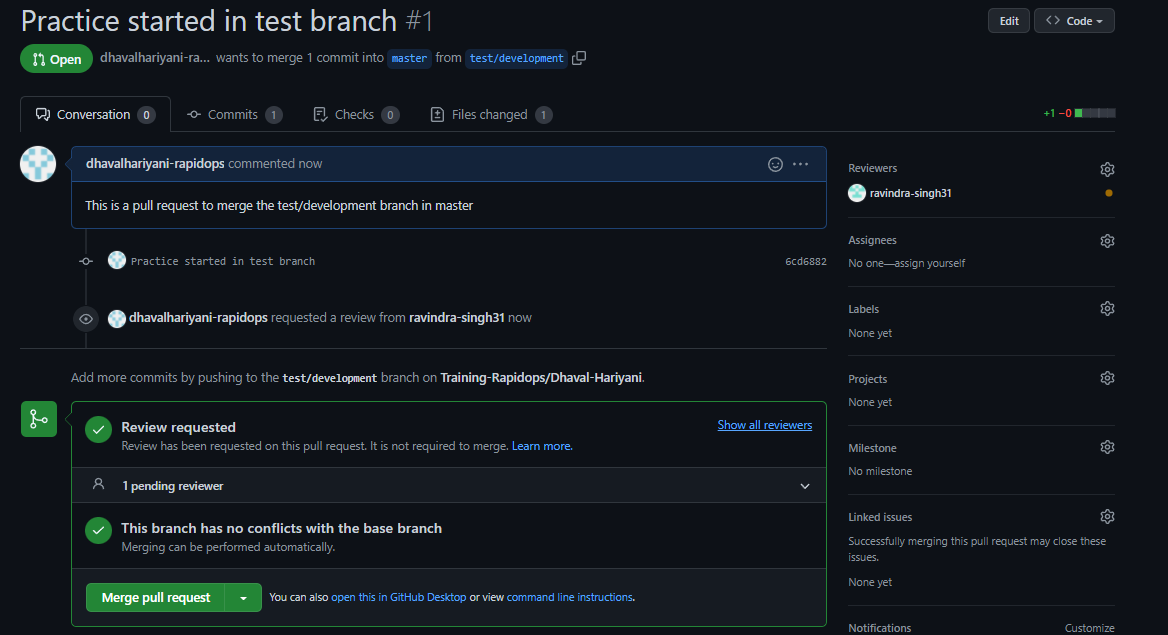




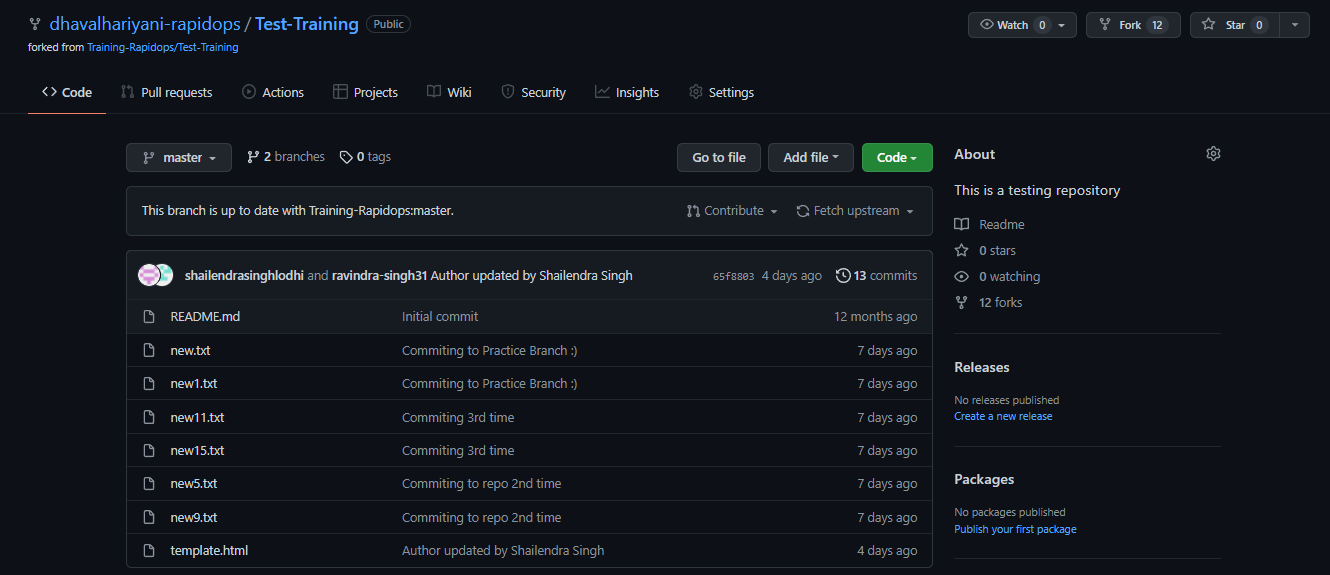


1. **Go to your GitHub repository and create a pull request to merge this branch in master. Also, add Ravindra & your mentor as reviewers.**



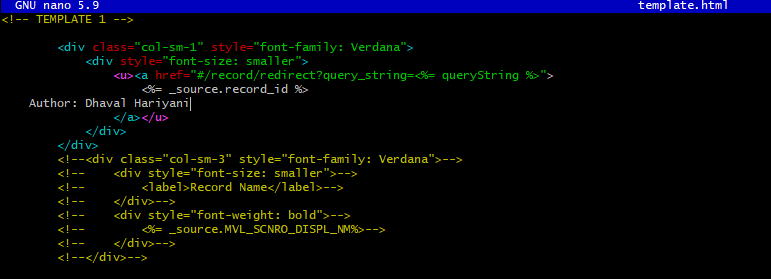


1. **Fork a public repository** [**Test-Training**](https://github.com/Training-Rapidops/Test-Training)

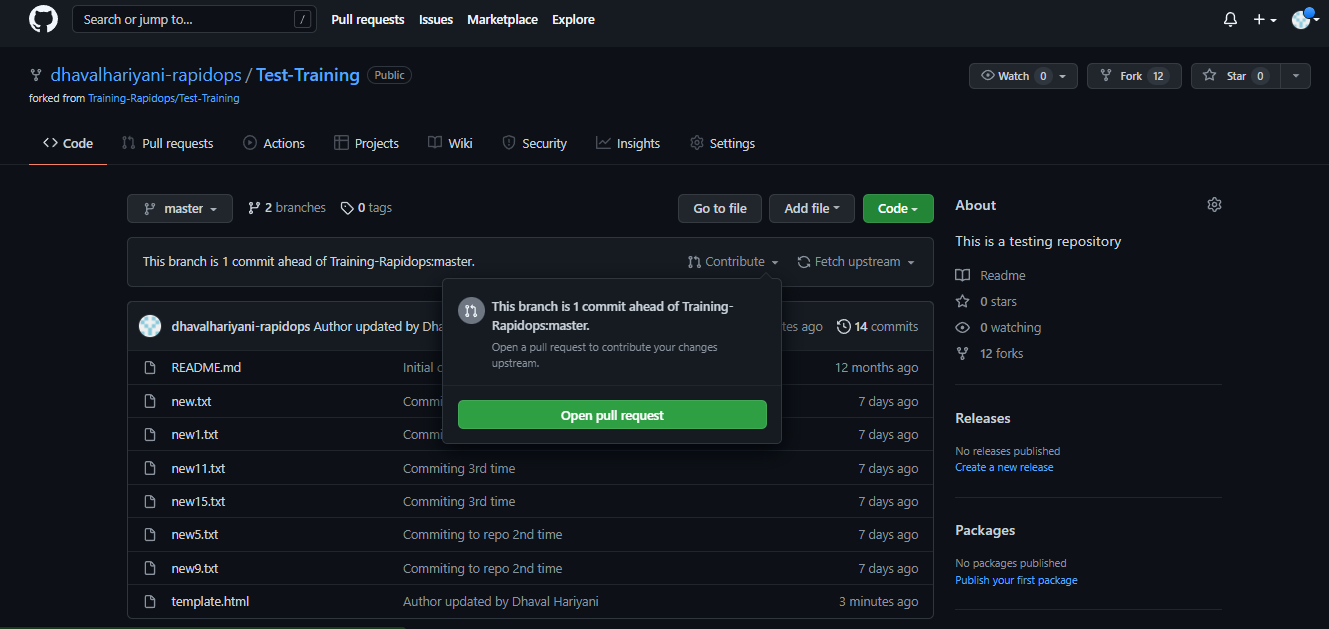


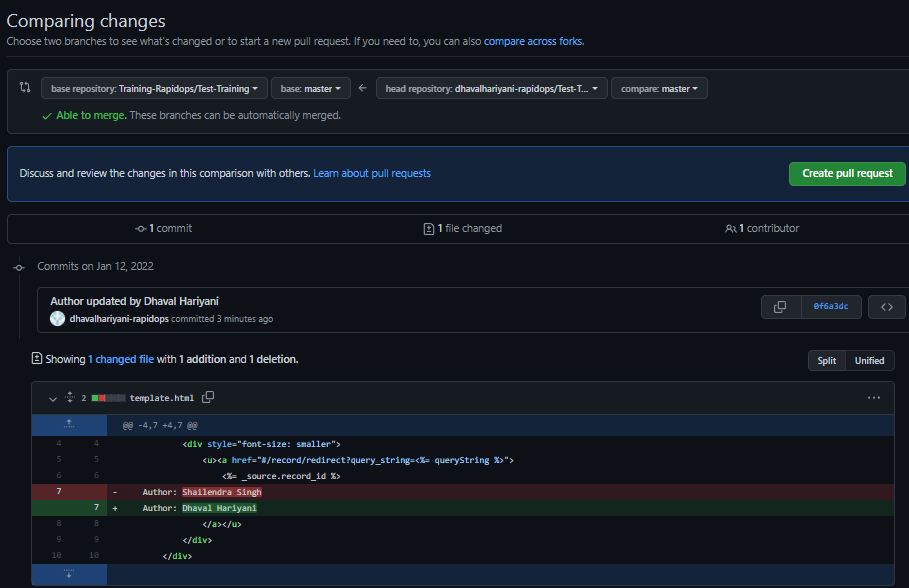
1. **Make a change in the template.html file by adding 'author: {your-name}' and add commit 'Author updated by {your-name}'.**

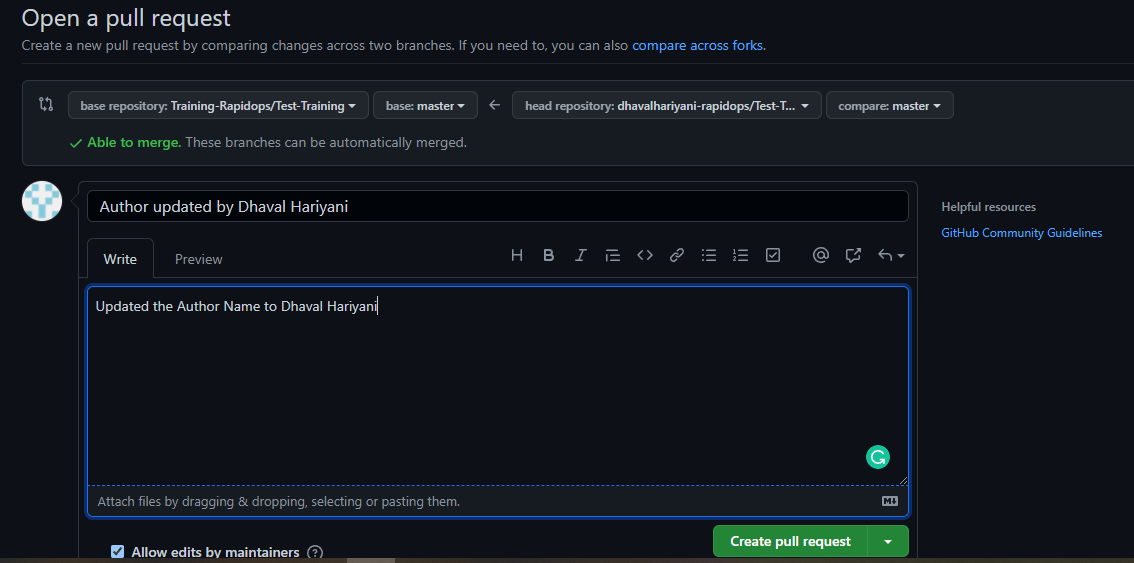
nano template.html

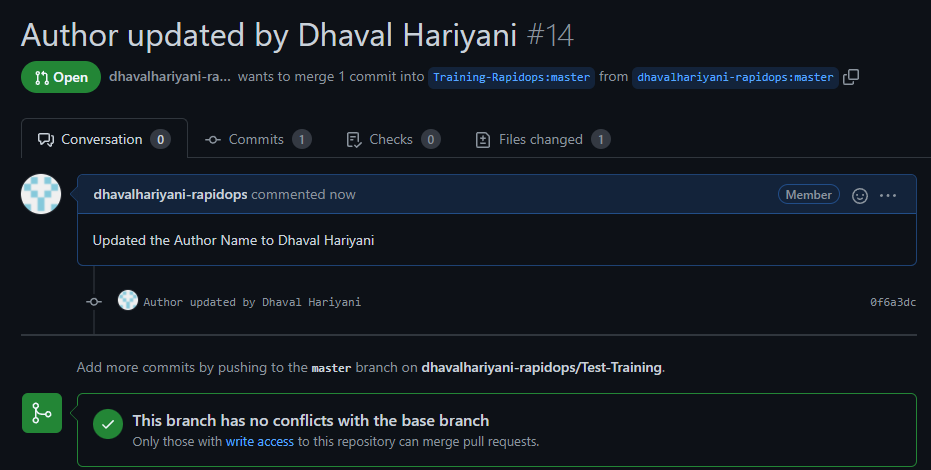


1. **Give PR to merge your forked repo in the Test-Training repo.**









1. **Come back to your repo, where you create a branch 'practice' and add 3 commits then rebase it with the master.**

git checkout -b practice

echo "First File" >> f1.txt

git add f1.txt

git commit -m "first commit"

echo "second change" >> f1.txt

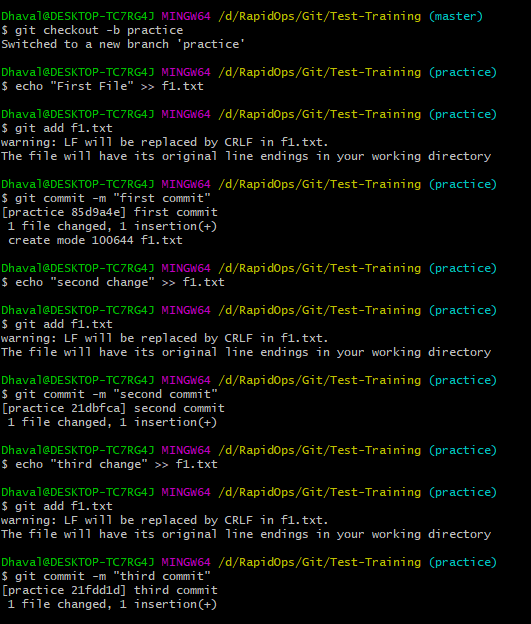
git add f1.txt

git commit -m "second commit"

echo "third change" >> f1.txt

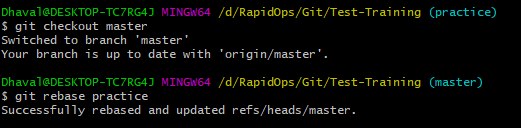
git add f1.txt

git commit -m "third commit"



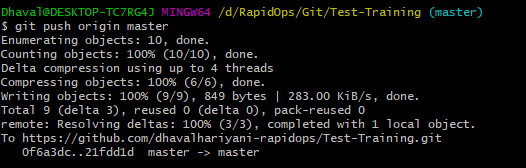
git checkout master

git rebase practice



1. **Push your changes to remote, then Add 3 commits again and squash them into the first commit by keeping the message "Rebase squash done".**

git push origin master



echo "Second File" >> f2.txt

git add f2.txt

git commit -m "first commit for F2"

echo "second change" >> f2.txt

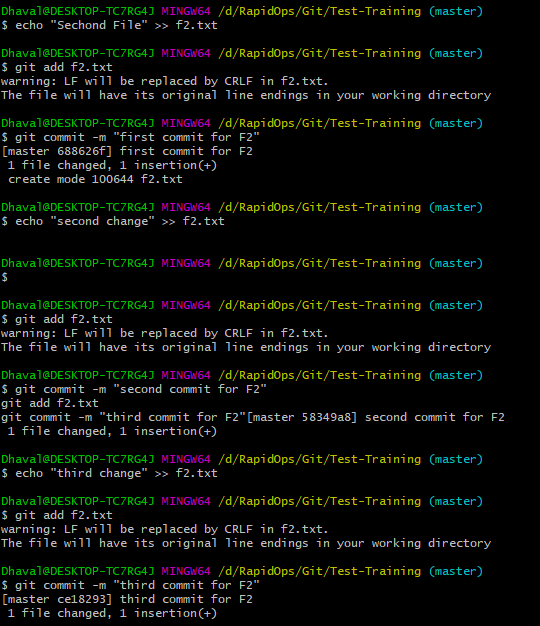
git add f2.txt

git commit -m "second commit for F2"

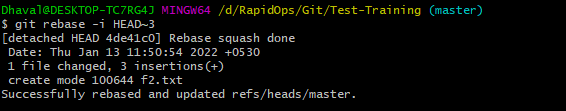
echo "second change" >> f2.txt

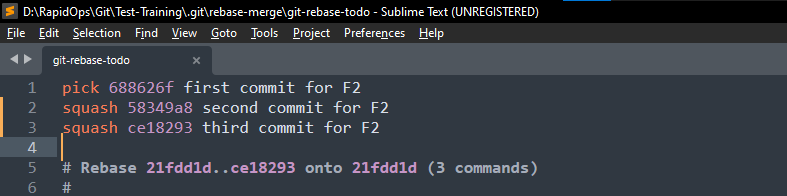
git add f2.txt

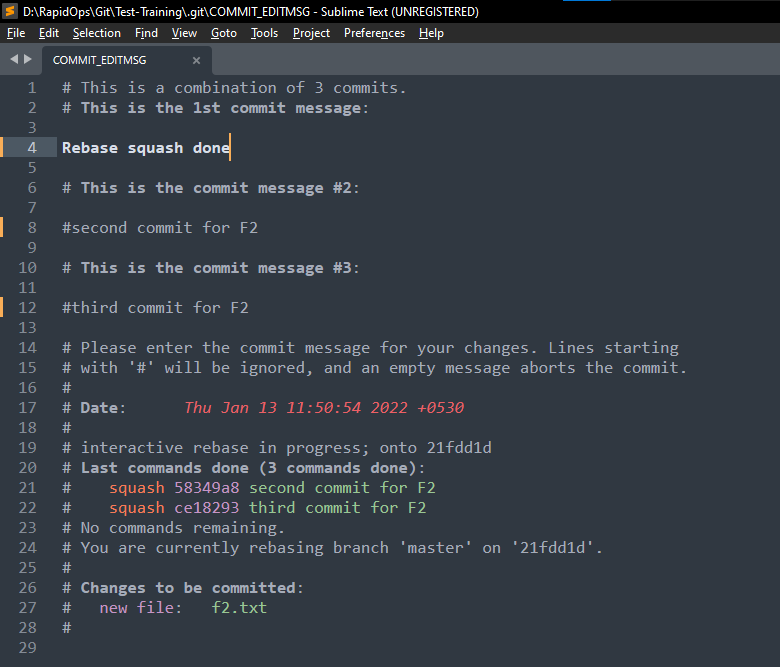
git commit -m "third commit for F2"



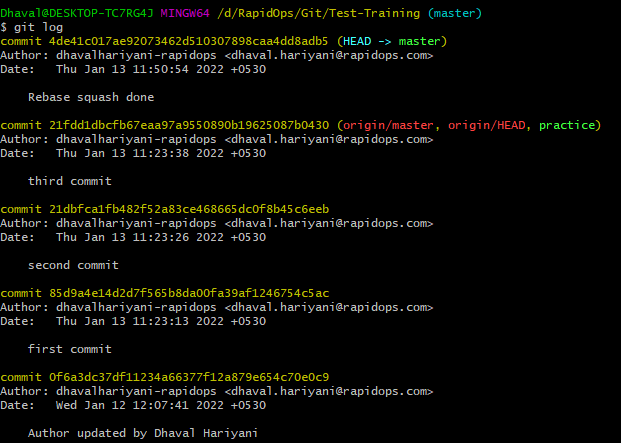
git rebase -i HEAD~3





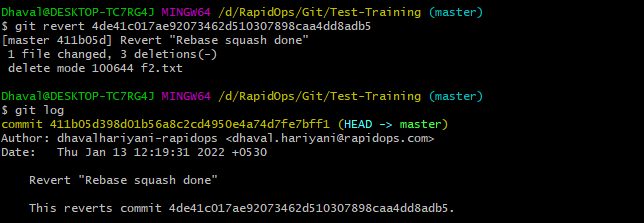


git log



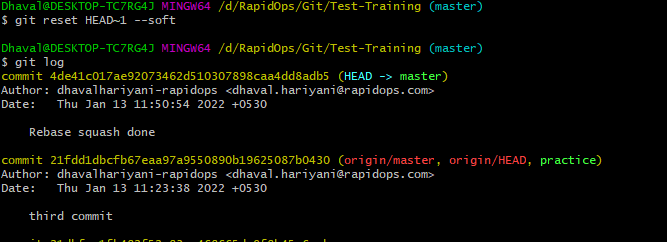
1. **Now revert these changes but note that the changes must be retained in the commit history. (****use default revert commit message)**

git revert 4de41c017ae92073462d510307898caa4dd8adb5



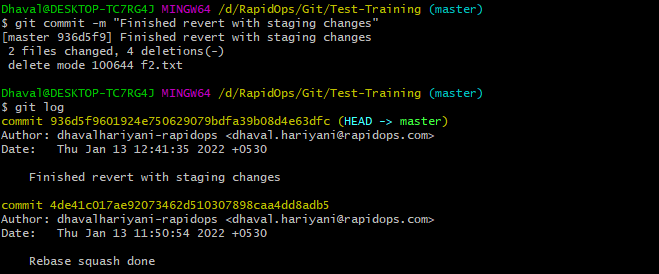
1. **Reset your last commit without losing the changes and then commit with the message "Finished revert with staging changes".**

git reset HEAD~1 –soft



git commit –m “Finished revert with staging changes”

git log



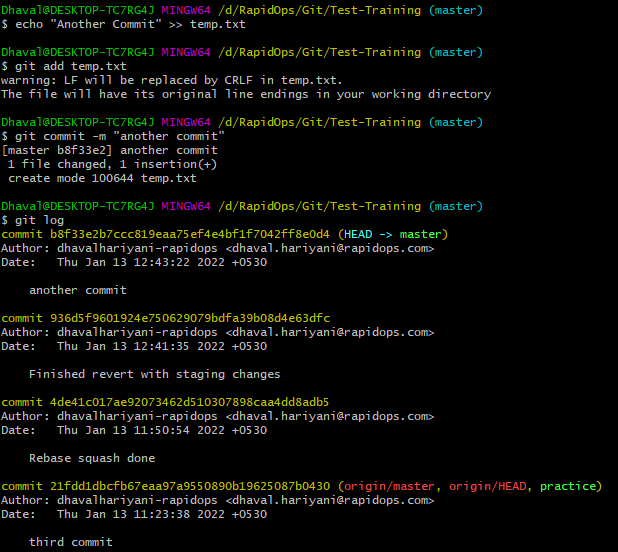
1. **Create one commit then perform a hard reset such that you're back to the commit with the message "Rebase squash done".**

echo “Another Commit” >>temp.txt

git add temp.txt

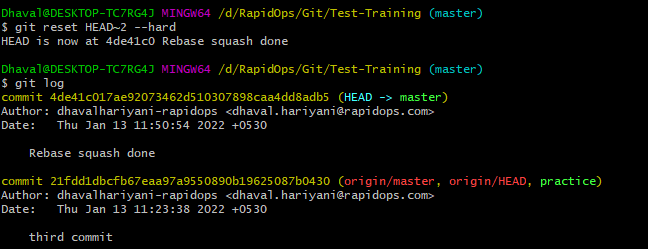
git commit –m “another commit”

git log



git reset HEAD~2 –hard

git log



1. **Create a file index.html, and add it to the staging index then stash it.**

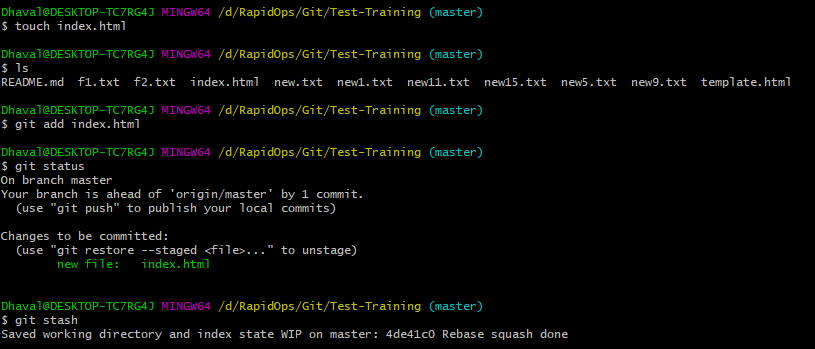
touch index.html

ls

Git add index.html

git status

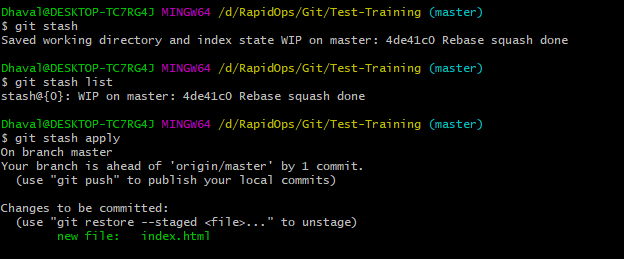
git stash



1. **Check the list of** **stash, what changes are there in the stash, then bring your changes from stash.**

git stash list

git stash apply



1. **Commit with a message "Revert, Reset & Stash done".**

git commit –m "Revert, Reset & Stash done"

