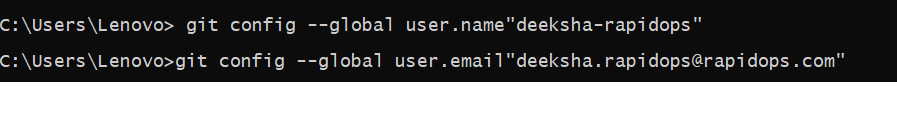
1. Configure your user name and email.



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

cmd-> git clone “address”

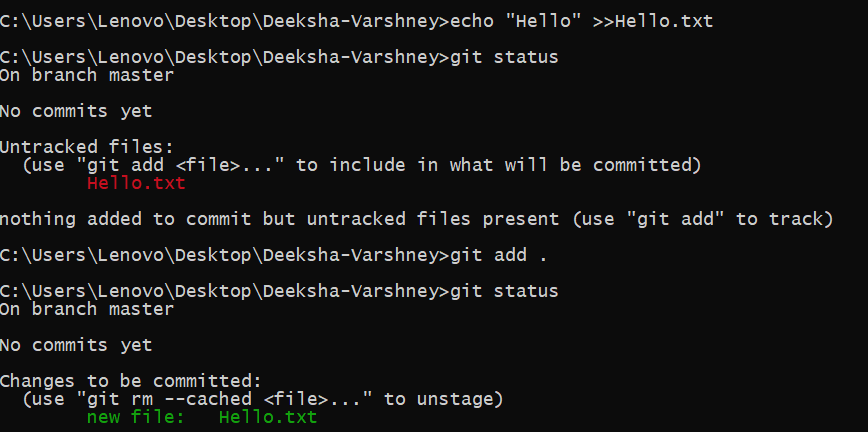


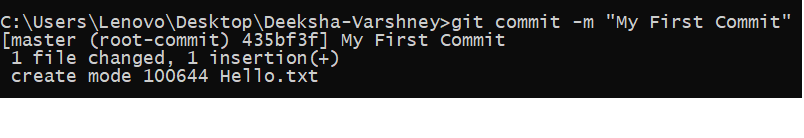
1. Create a file inside the repo, and make your first commit "My First Commit".

cmd-> echo “hello”>>file name

git add .

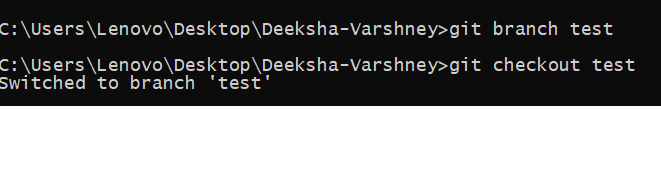
Git commit -m “My first commit”





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

cmd-> git branch test , git checkout test

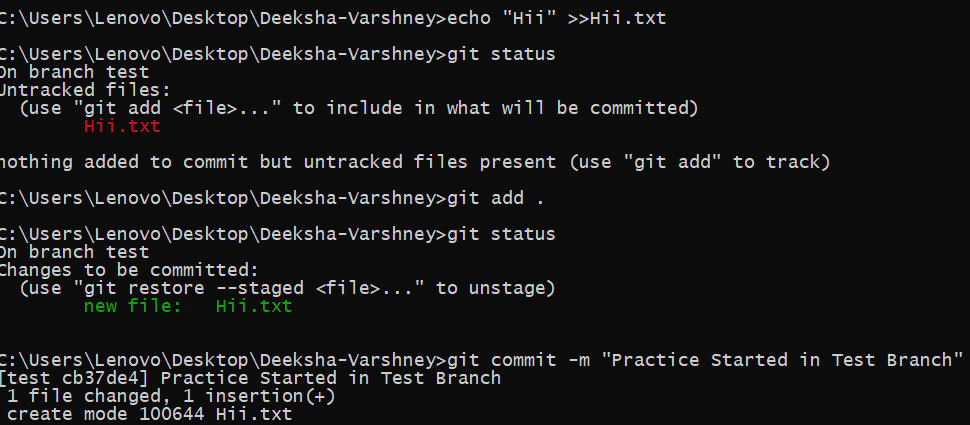


1. Add a file in this branch and commit your changes with the message "Practice started in test branch".

cmd-> echo”Hii” >>Hii.txt

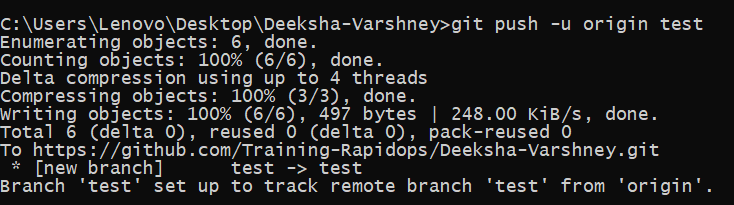
git add .

git commit -m ”Practice Started in Test Branch”



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

Cmd -> git push -u origin test

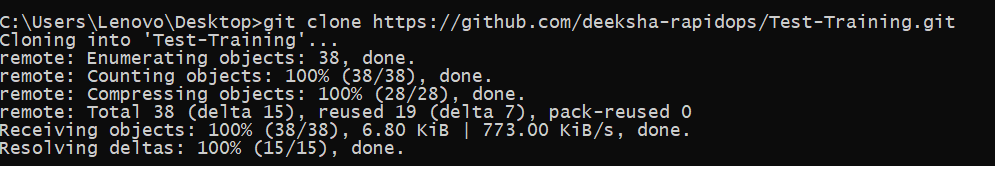


1. Go to your GitHub repository and create a pull request to merge this branch in master. Also, add Ravindra & your mentor as reviewers.

Created pull request and merge branch.

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

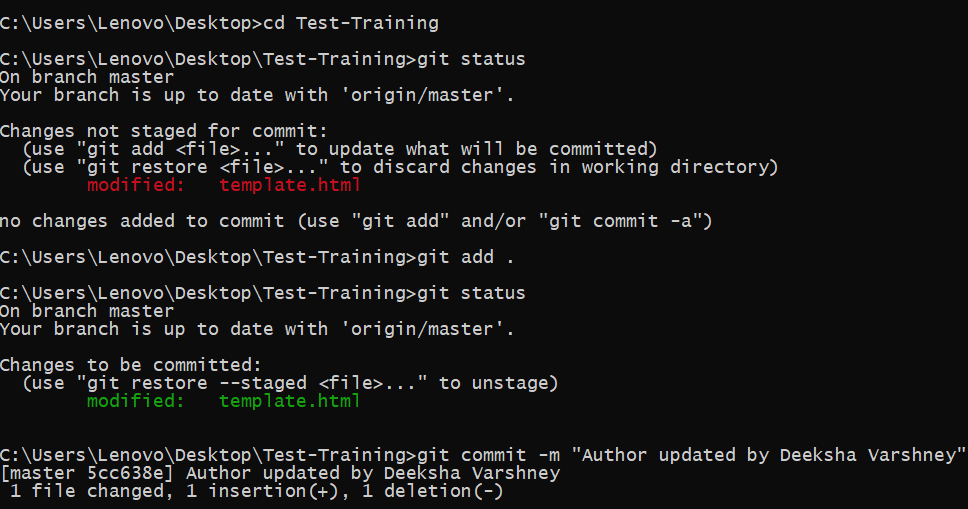
Forked in github.



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

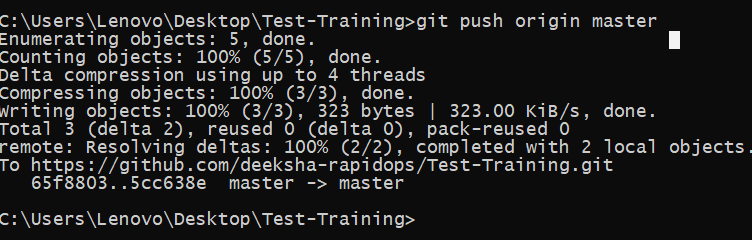
Cmd-> git add .

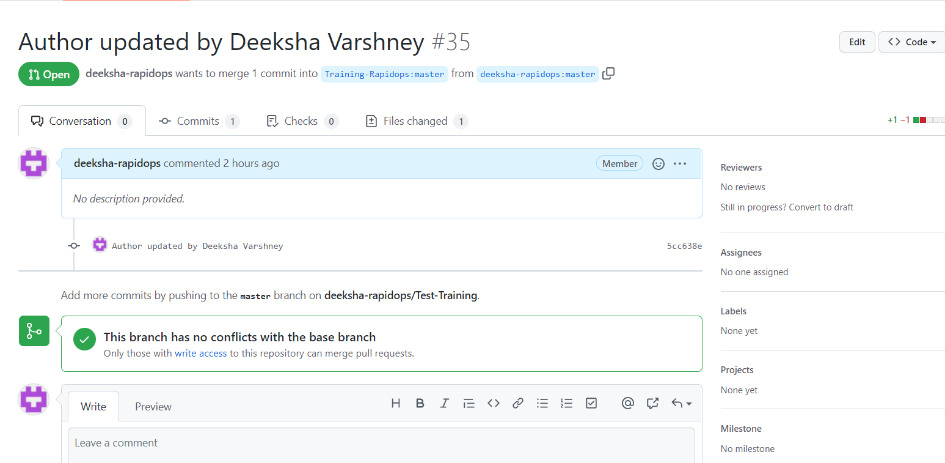
git commit -m “Author updated by Deeksha Varshney”



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

Made a pull request on github.





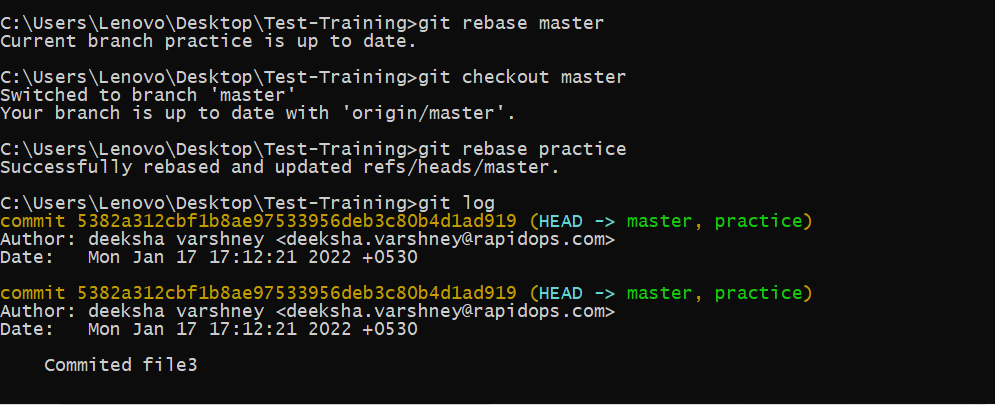
1. Come back to your repo, where you create a branch 'practice' and add 3 commits then rebase it with the master.

Cmd-> git branch practice , git checkout practice

Echo for 3 file creations

Git commit for commit each file(Missed screenshots of these commands)

All 3 commits are rebased with master.

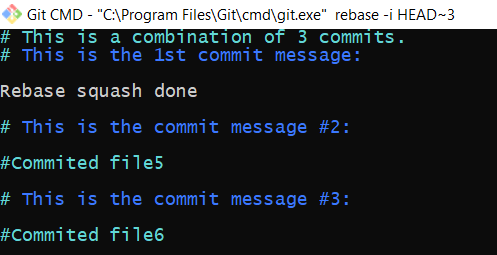


1. Push your changes to remote, then Add 3 commits again and squash them into the first commit by keeping the message "Rebase squash done".

Cmd-> git push origin master(Missed Screeshots)

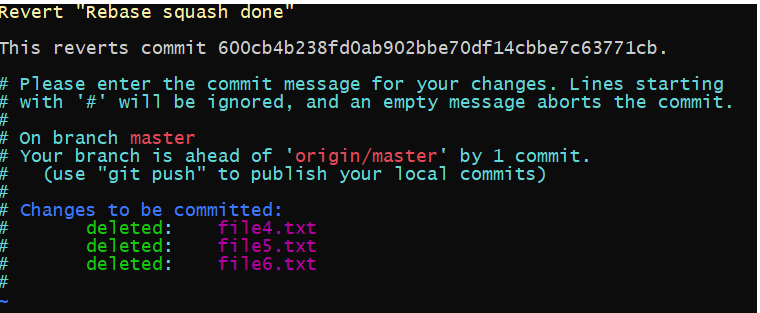
Cmd-> git rebase -I HEAD~3

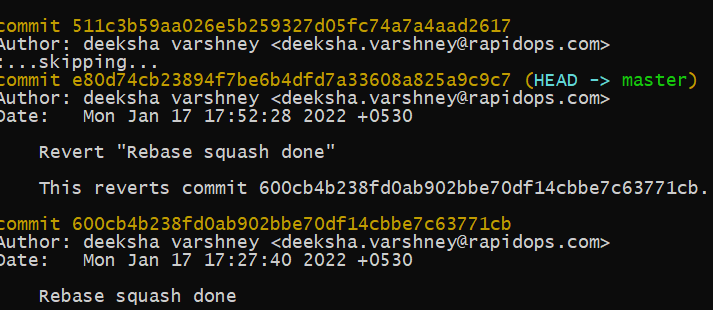
Rebase squash done



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

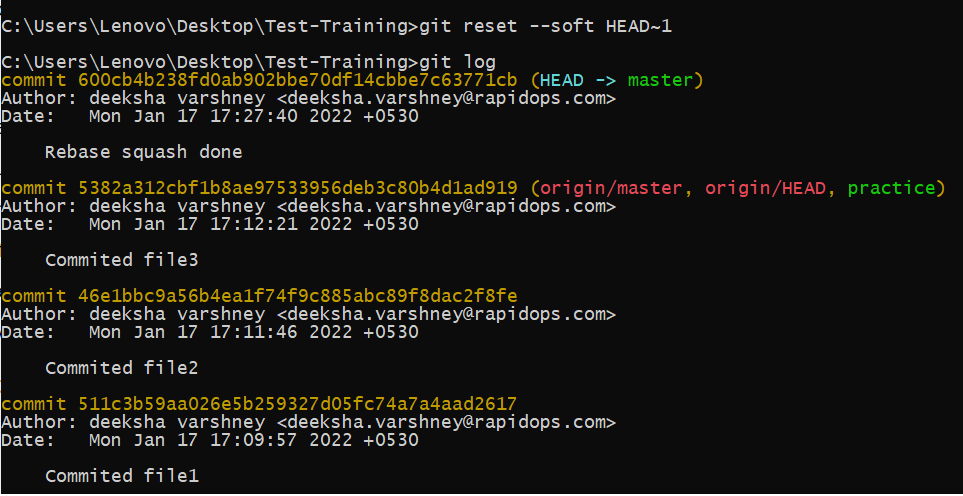
Cmd-> git revert “commit id”





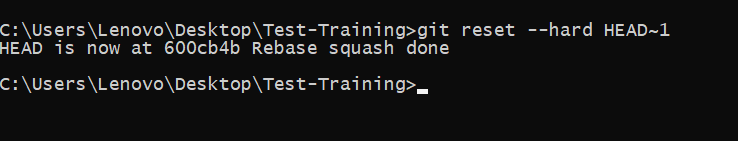
1. Reset your last commit without losing the changes and then commit with the message "Finished revert with staging changes".

Cmd-> git reset –soft HEAD~1



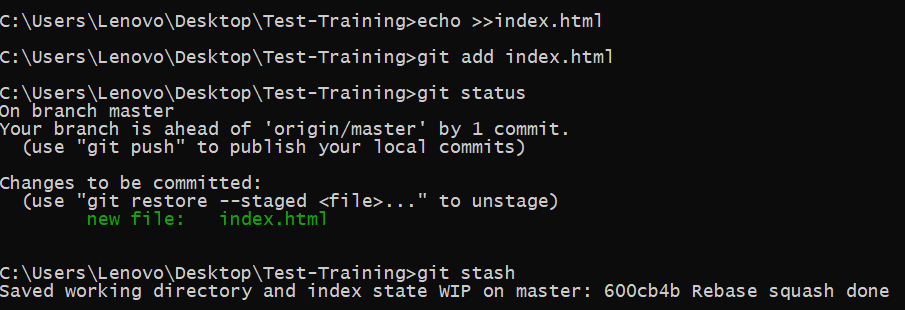
1. Create one commit then perform a hard reset such that you're back to the commit with the message "Rebase squash done".

cmd-> git reset –hard HEAD~1



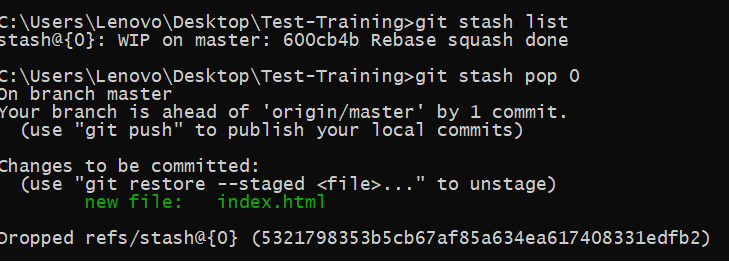
1. Create a file index.html, and add it to the staging index then stash it.

Cmd-> git stash



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

Cmd = git stash list , git stash pop 0



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

