Let’s learn about Github branching commands

What is an Environment

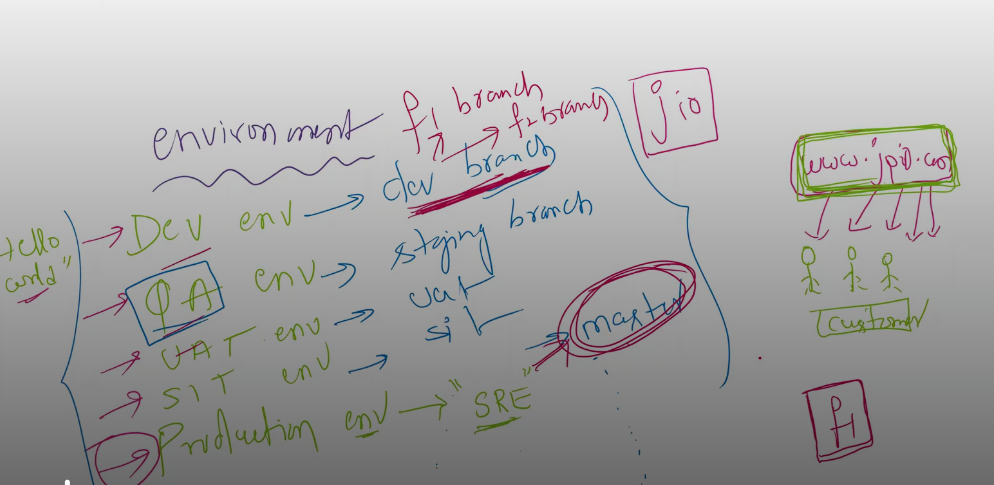
Environment is nothing but a Location where the code is stored

Below is the Types of Environment with their Respective branching

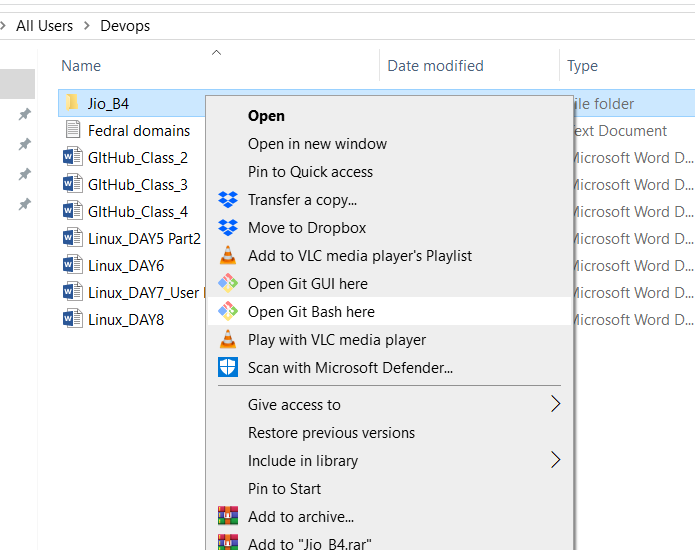
Dev Environment ------ dev branch

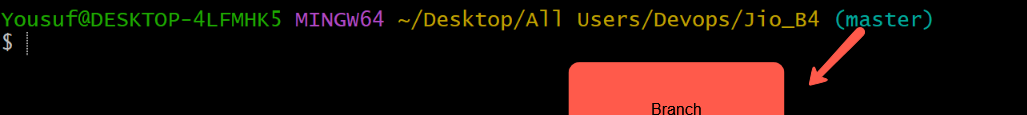
(QA)Testing Environment[staging Environment] ----- staging branch

Production Environment ---

When a request is sent to a developer to update the login page, the developer creates a new branch(f1 branch,f2branch) from the existing (dev branch)master branch, where the current code is stored. The existing code from the master branch is copied into this newly created branch, allowing the developer to start working on the updates separately.  
  


The default branch would be Master branch . Let’s connect to Git bash and check it



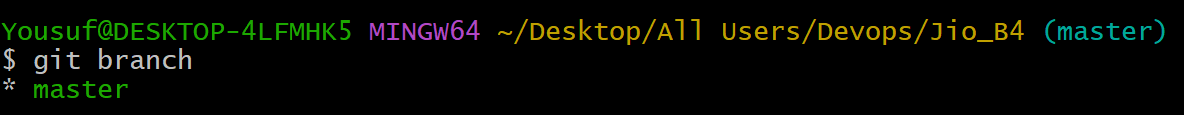


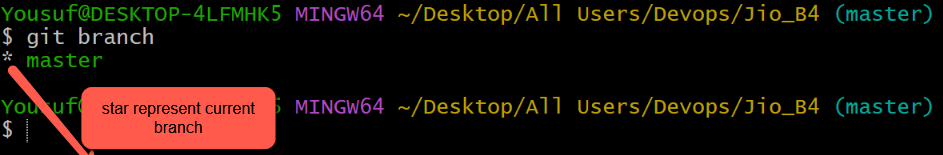
Git branch command

The git branch command lets you manage branches in your Git repository. Branches help you keep your work separate and organized.

To identify the no of branches we have to use this command git branch

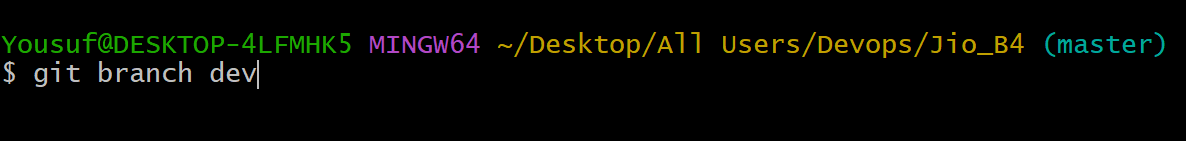
Git branch

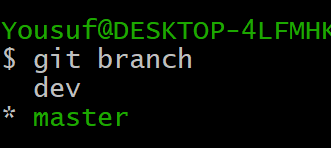
As you can see there is only one branch present (ie) master branch  


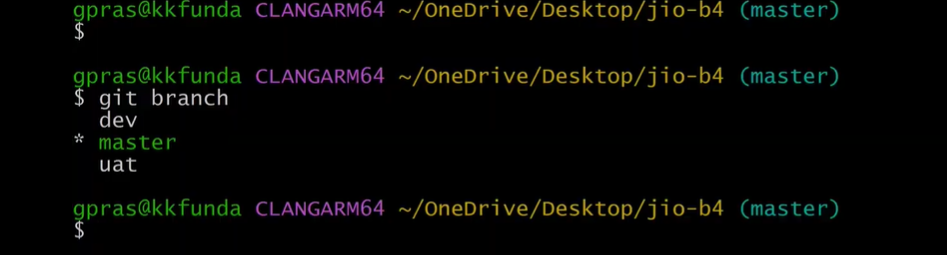


By using the command ‘git branch’ we can create the new branch , we can list the branches , we can delete a branch , and we can rename the branch.

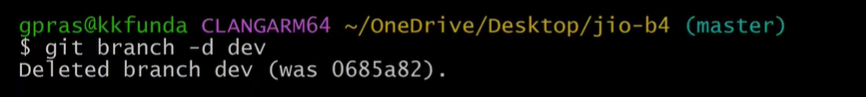
Let’s create a branch dev using the command git branch command



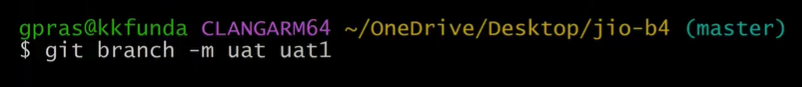




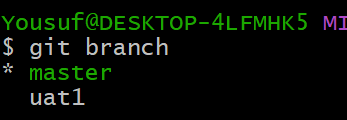
To delete a branch we have to use the command “git branch –d dev “



To rename the branch we have to use the command ‘git branch –m old branch name new branch name’



24:03



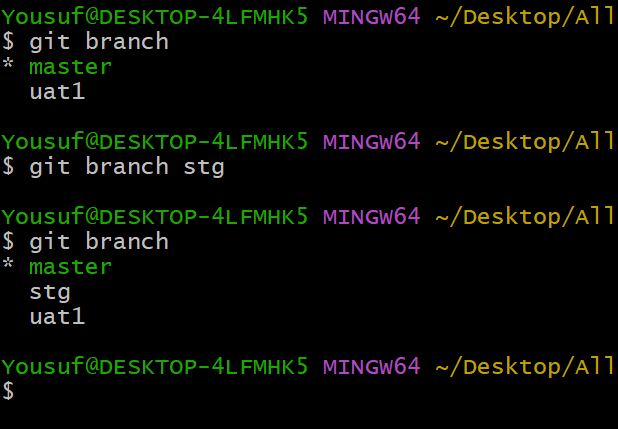
There are 7 files in Jio\_B4 folder

Ls –lrth (to list the files)

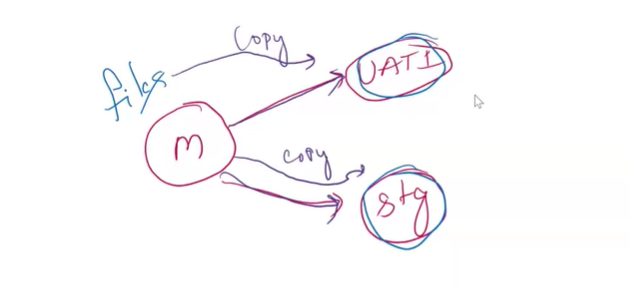
Ls –lrth|wc –l (to check no of files) it’s 7



Let’s create a new branch

Stg

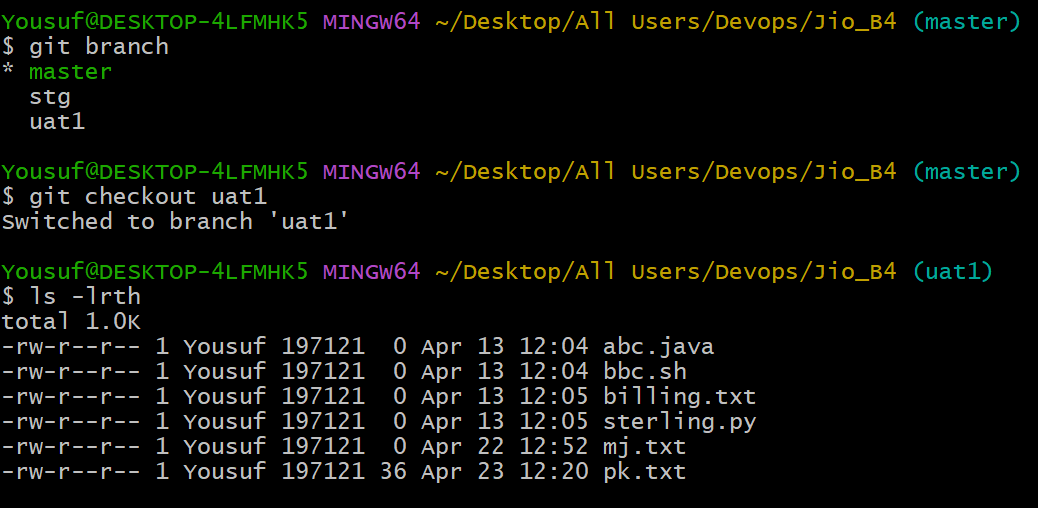
In the below image you can see we have created stg and uat1 branches from master branche so the files present in the master branch also get copied in to child branches(stg , uat1)

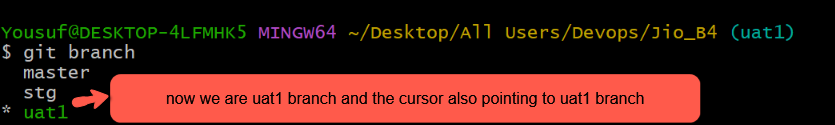


Let’s switch to child branches In order to make sure whether the files from master branch get copied to child branches uat1 , stg.

Command to switch from one branch to another branch

git checkout uat1

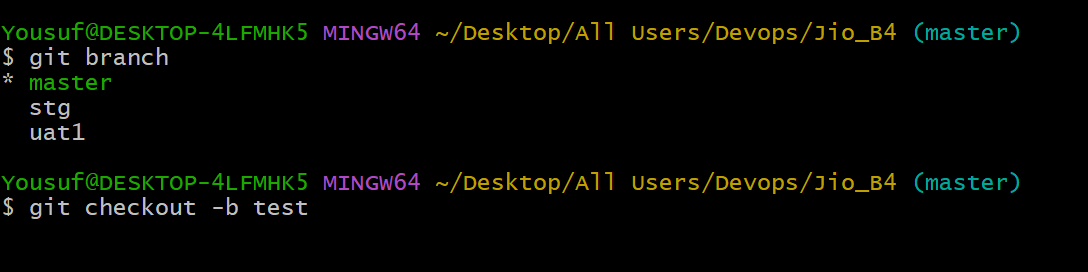


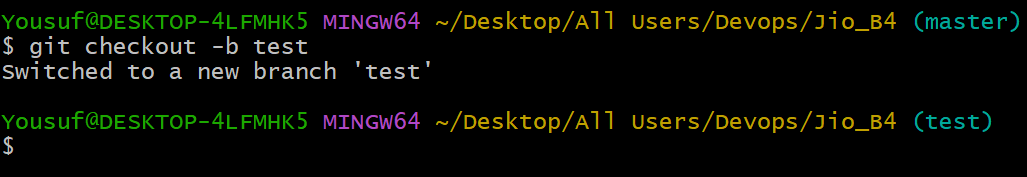


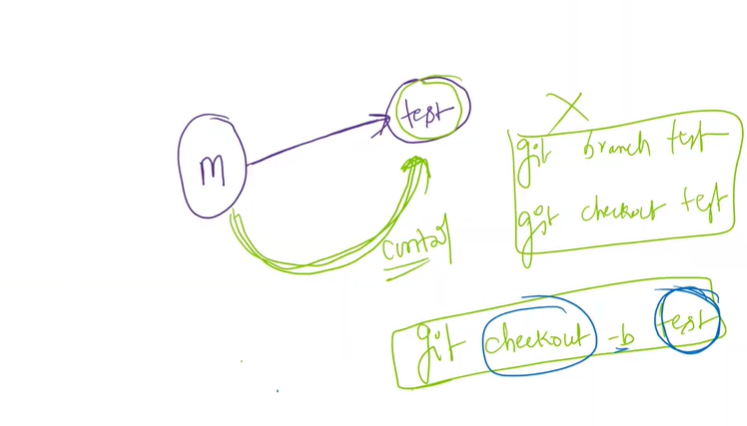
We don’t have to execute the commands git branch and git checkout separately

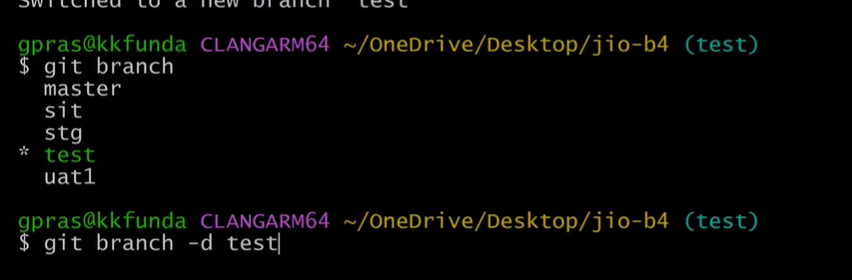
We can execute both commands at a same time

git checkout –b test  
  
This command will create a test branch with the help of –b & then it will automatically switch to the test branch.

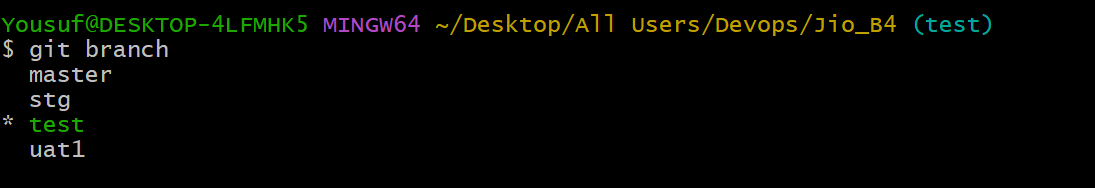




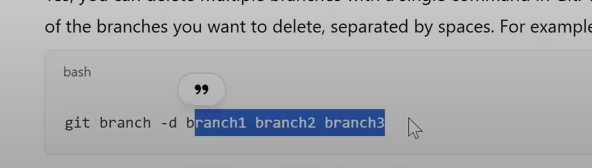


prashant 

Here in the above image you can see that we cannot directly delete the current working branch(test) .so to delete it we have to move to another branch from another branch we can delete the test branch



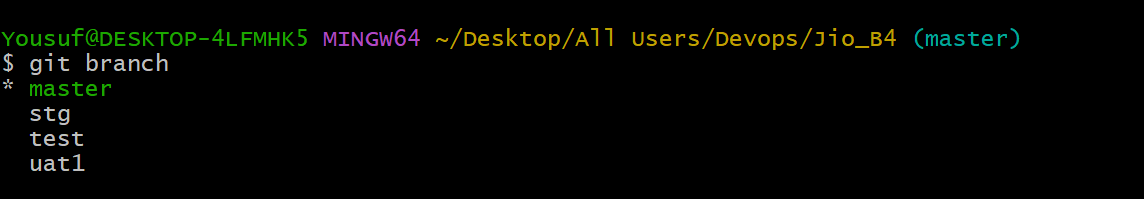


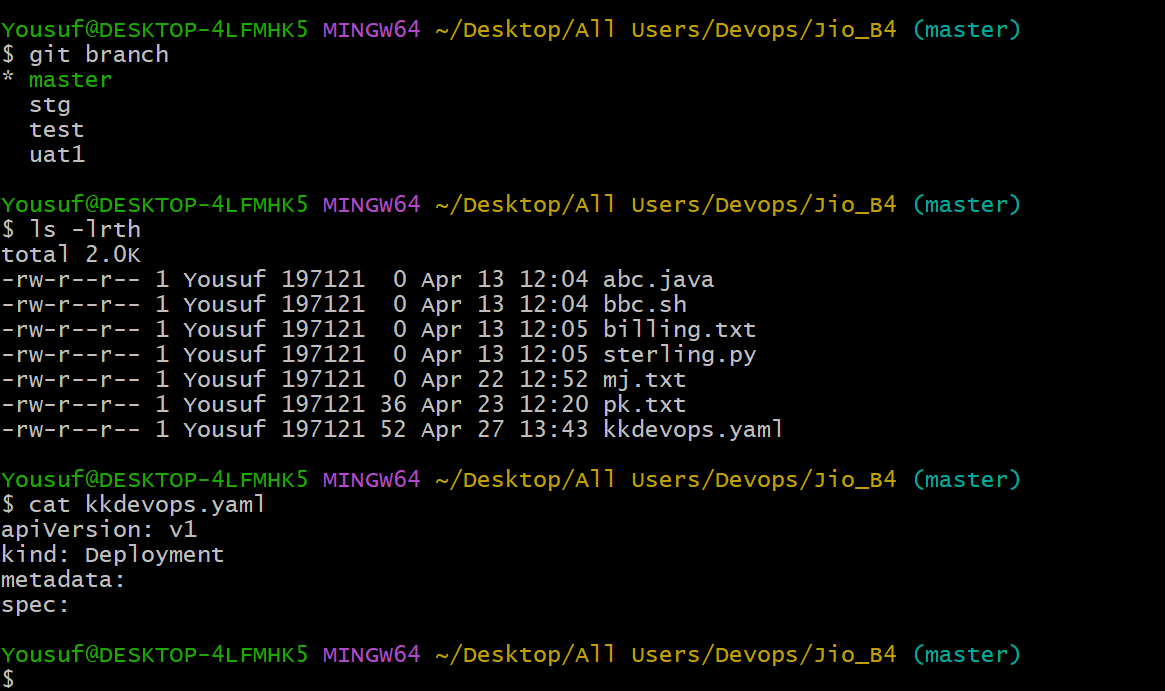
We can delete multiple branches at a time using the below command  


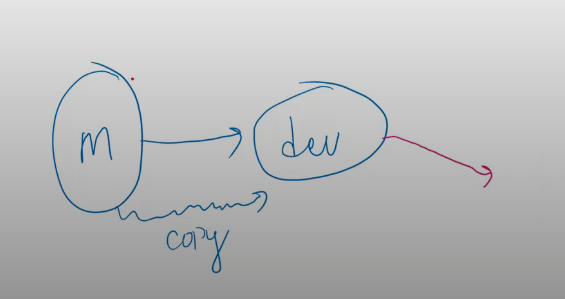
**Branch Merging:**

Lets see how to merge one branch code to another branch

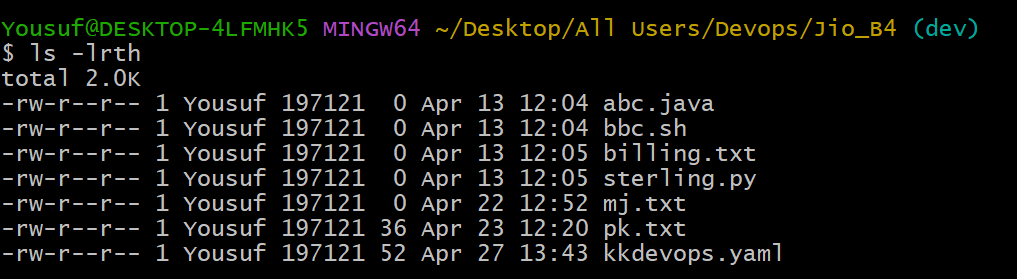
Note : don’t do the blind merges

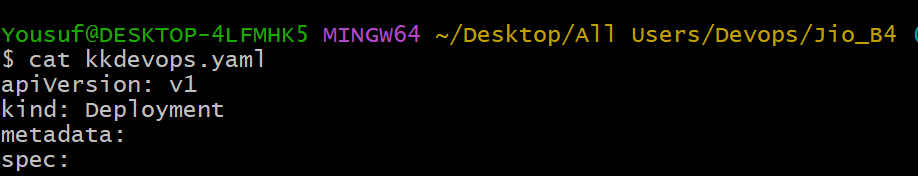
First check the available files in the master branch  


In the master we have a file called kkdevops.yaml  


All the files present in master branch also get copied to its child branch   
  
Now switch to dev branch  


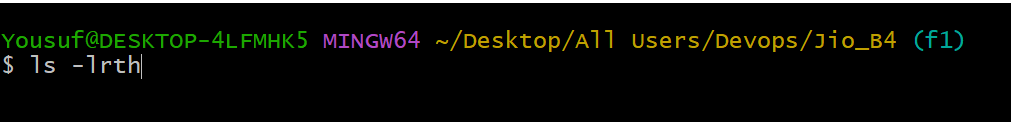


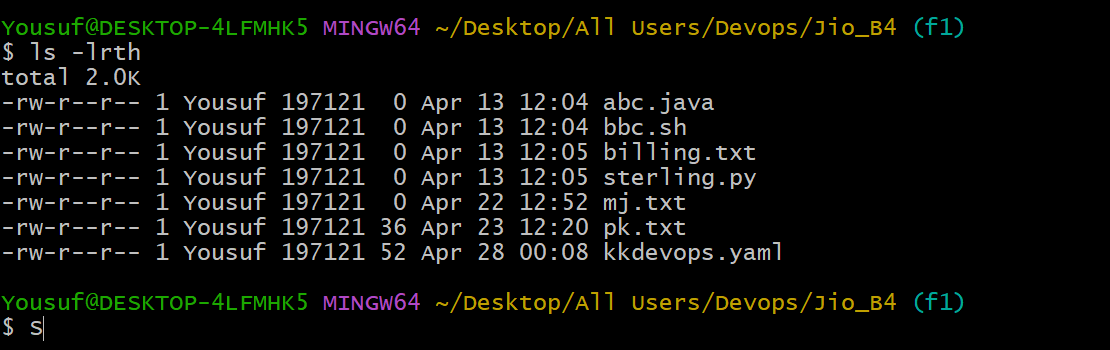
Ls –lrth (to check the list of files copied from master branch)  


Open kkdevops.yaml  


47:00 min

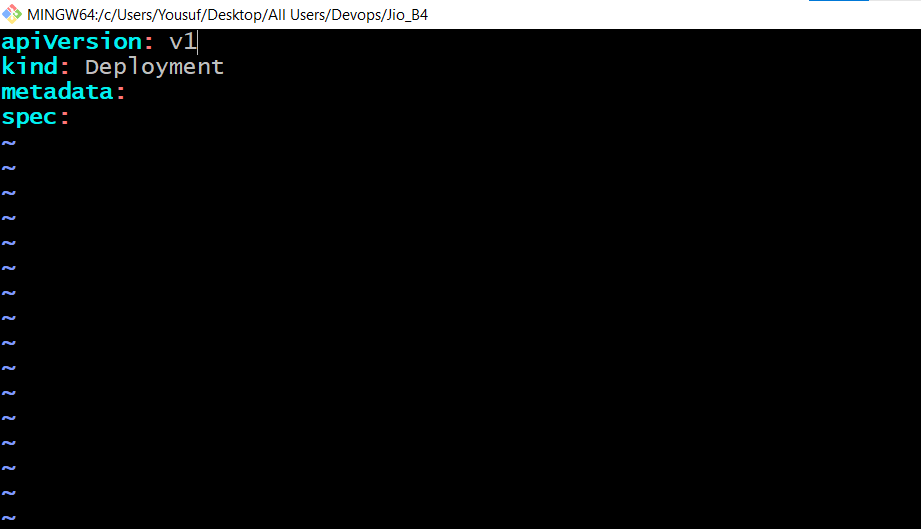
Now create a branch from f1 under dev branch and modifiy the kkdevops.yaml file in f1 and try to merge the same code in dev

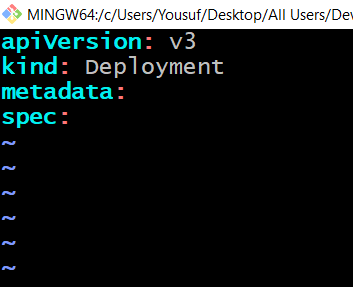




Open kkdevops.yaml and modified it change the version from v1 to v3

Whenever we merge the code from one branch to another branch don’t do the blind merges

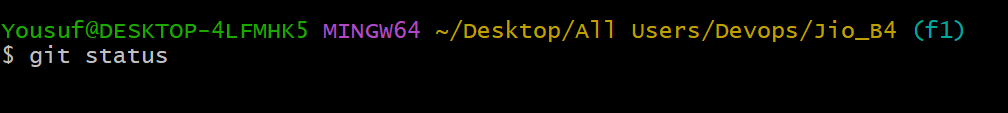


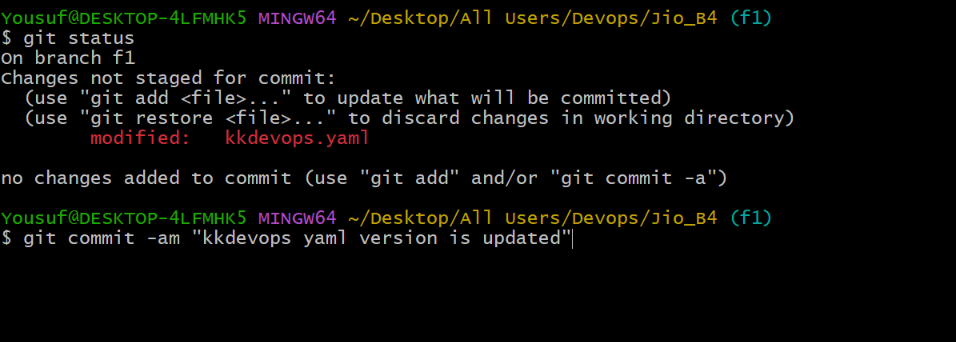
Change the version to v3 and save it  


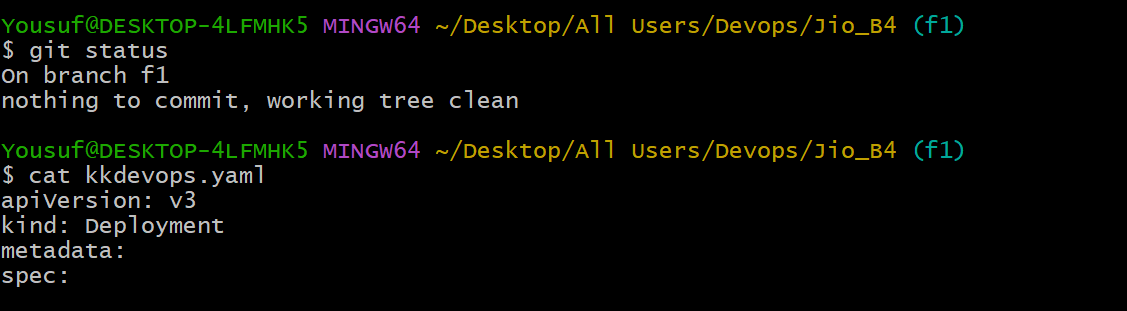
Now don’t directly merge the code present in f1 branch to dev branch

first we need to commit the updated code of kkdevops.yaml of f1 branch and then merge in to dev branch

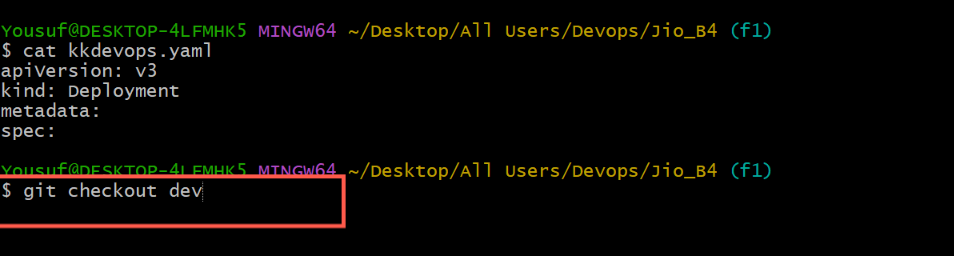
git status to check the modified code in kkdevops of f1 branch is commited or not

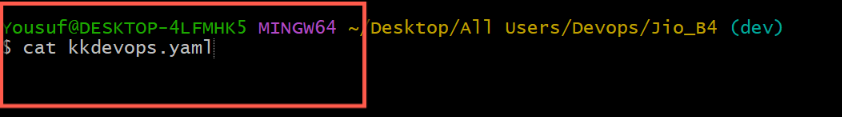






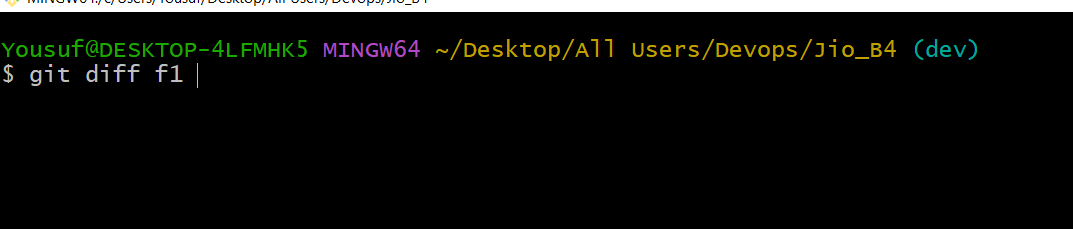
Now lets switch to dev branch and check the kkdevops code

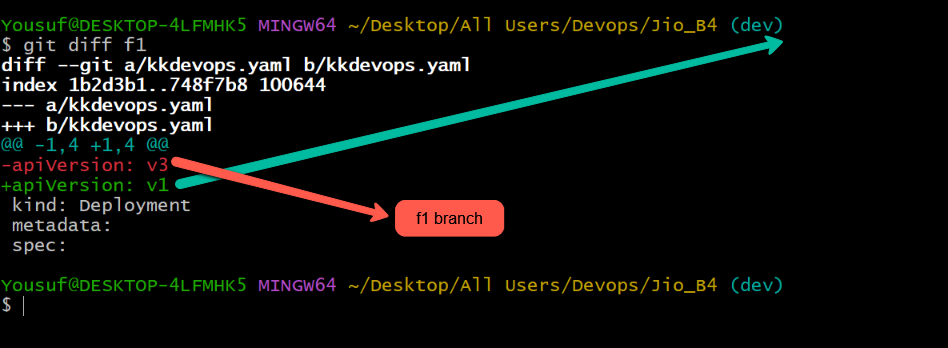




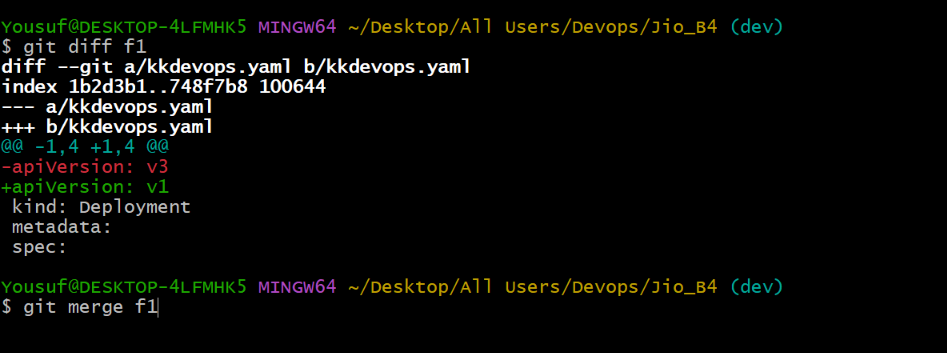


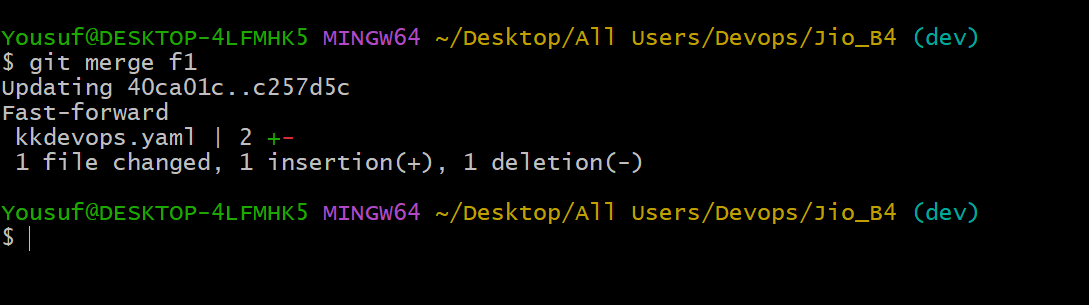
Before merging execute the command ‘git diff f1’ to check the difference of kkdevops file in dev and f1 branches



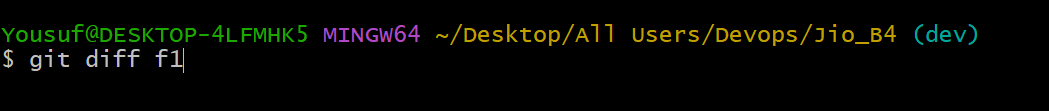


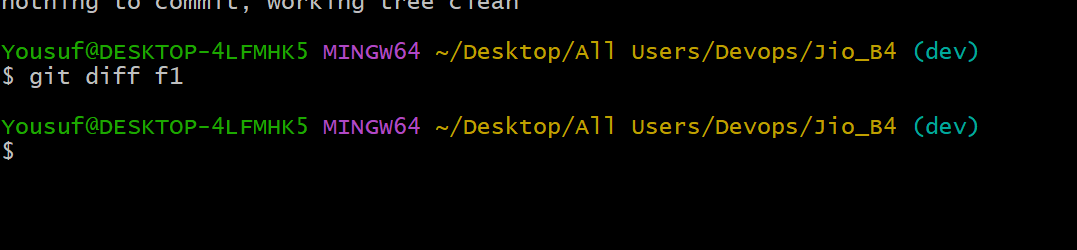
Now the controller is in dev branch so execute the merge command

git merge f1

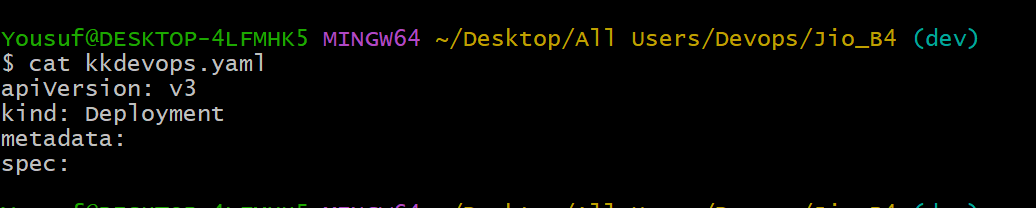


Git status

Now again execute the command git diff f1. If it does not show anything it means that file kkdevops.yaml from f1 is merged perfectly in kkdevops.yaml of dev branch. 

No Result from tfilehe below image  


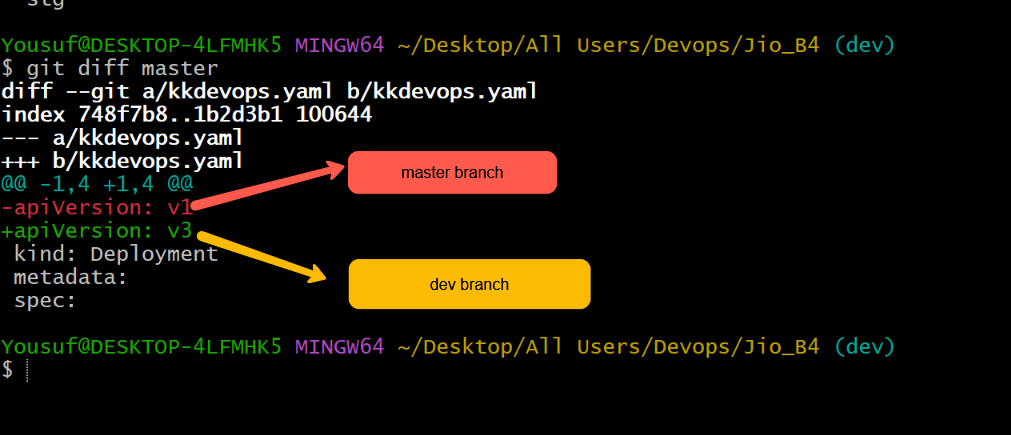
It means that f1 branch’s kkdevops.yaml file is perfectly merged in dev branch’s kkdevops.yaml



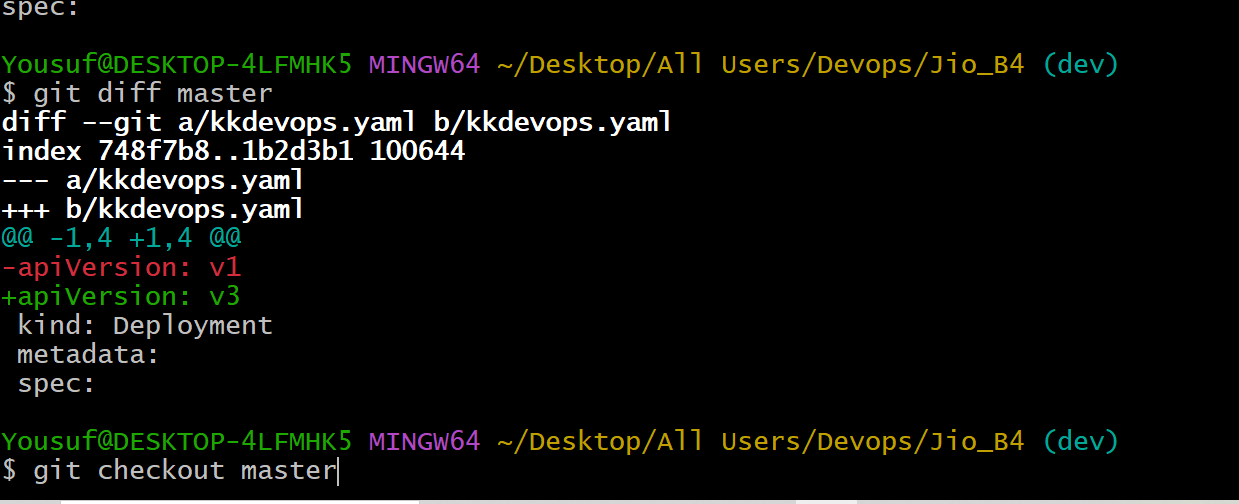
53:00

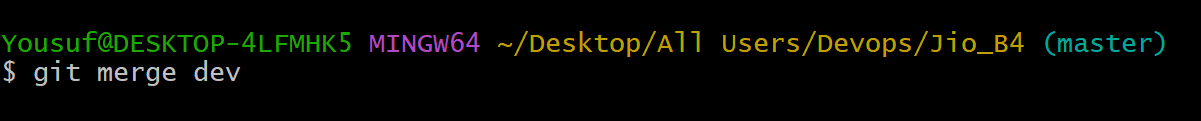
Now let’s merge the dev branch code in the parent branch master

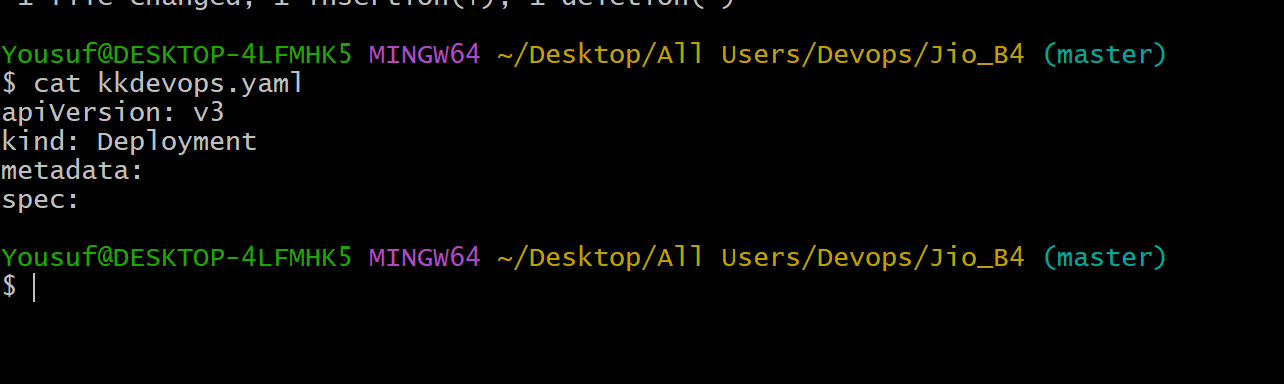
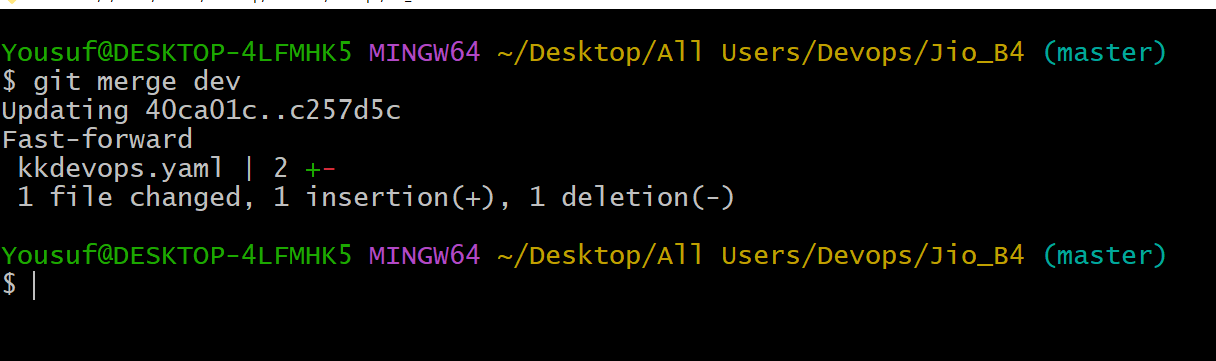
So first execute the git diff master in dev branch



Now move the cursor to the master branch and execute the command git merge dev







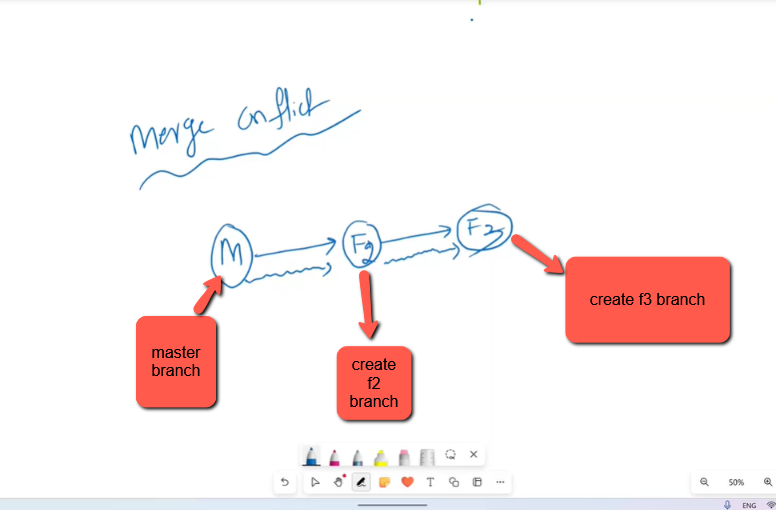
Merged successfully and code got update in master branch successfully

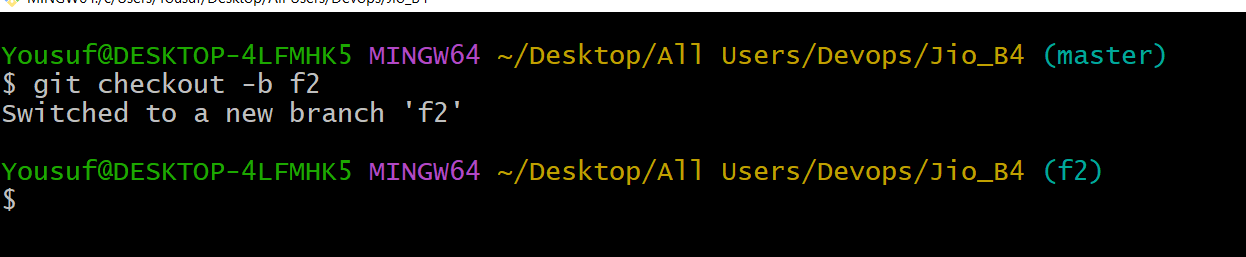
Now code in kkdevops.yaml in all the branches is up to date but some how in interview it will be ask about the merge conflict

What is merge conflict could you explain it ?

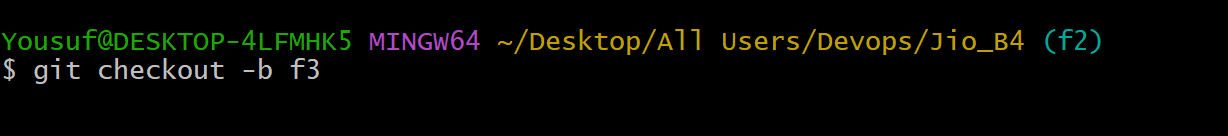
A **merge conflict** occurs in Git when two branches have modified the same file and same lines of code differently. Git can’t automatically decide which change should be kept, leading to a conflict. This typically happens when merging branches or pulling changes from a remote repository.

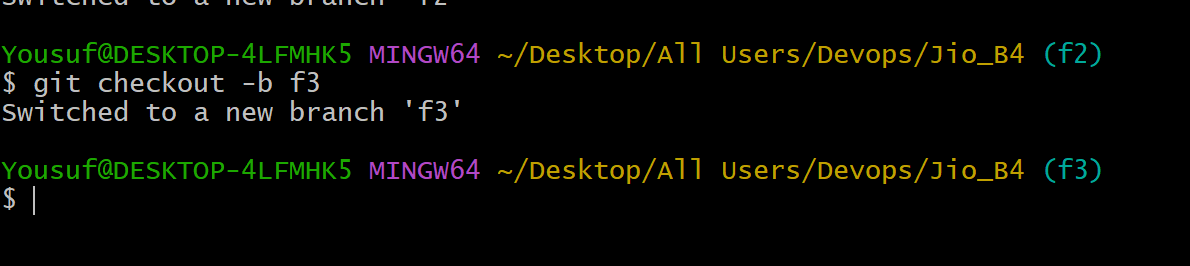
Scenario:



In master branch execute the command ( git checkout –b f2 ) 

From f2 branch Execute the command ( git checkout –b f3 )



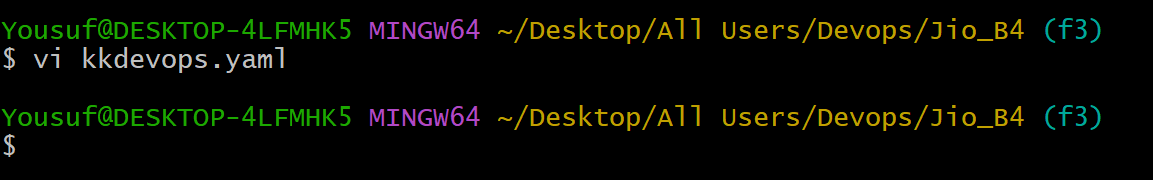


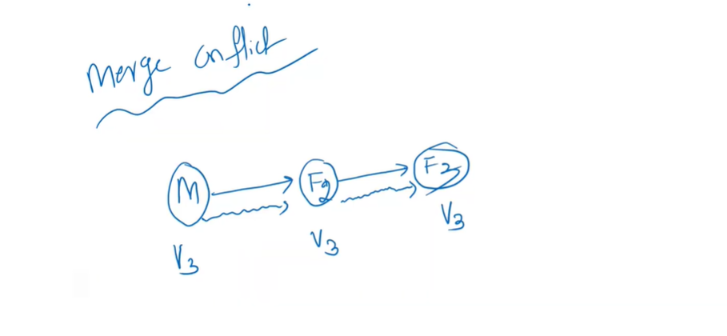
Let’s update the file kkdevops.yaml in f3 branch

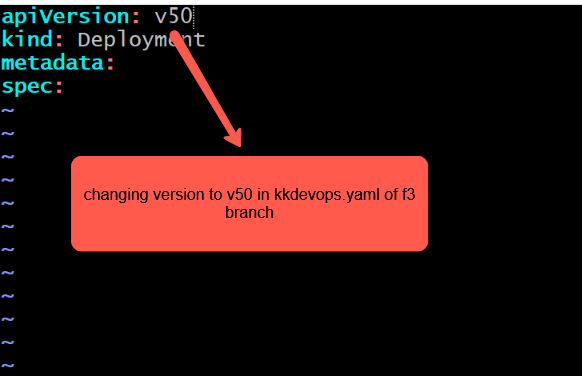
First open the kkdevops.yaml and check the version

execute the command ( cat kkdevops.yaml ) and check the version

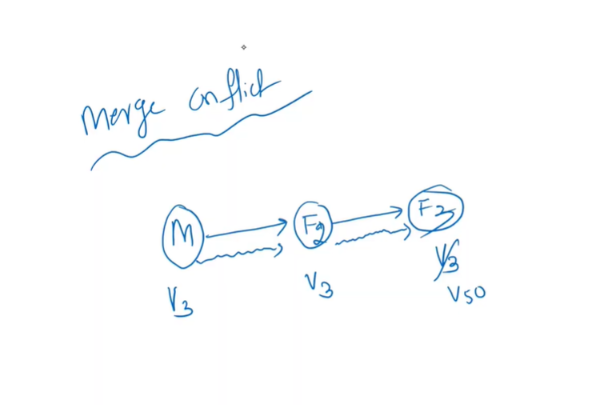


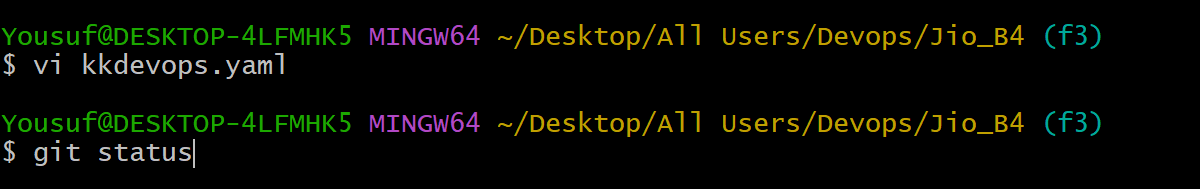


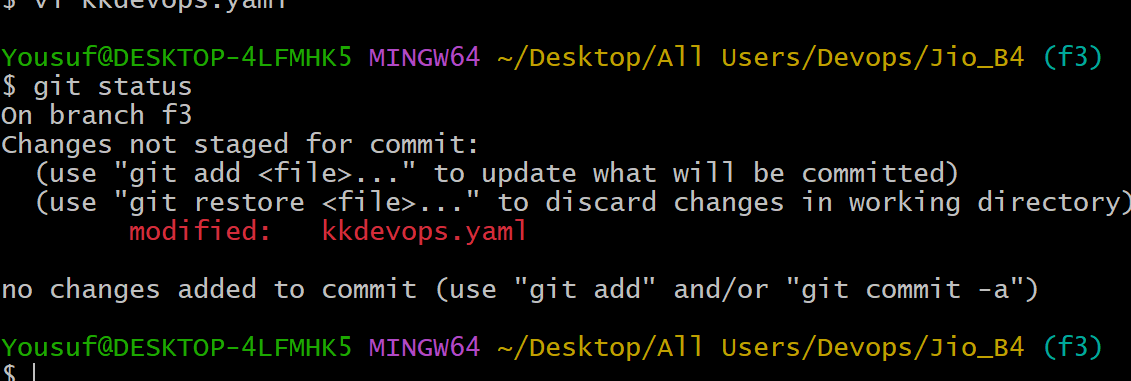
Now update the version or change the version to ‘v50’  
 present situation before updating the version to v50 in f3 branch  




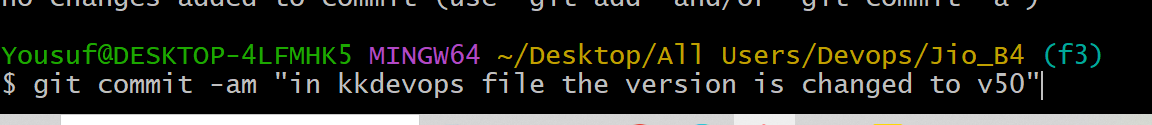
Save it and commit the modified code

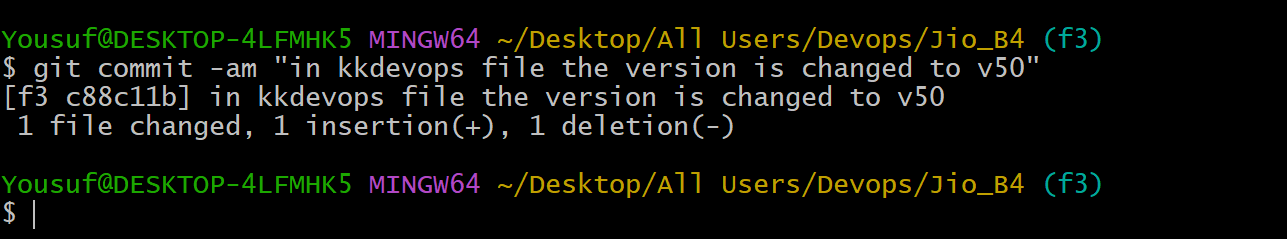




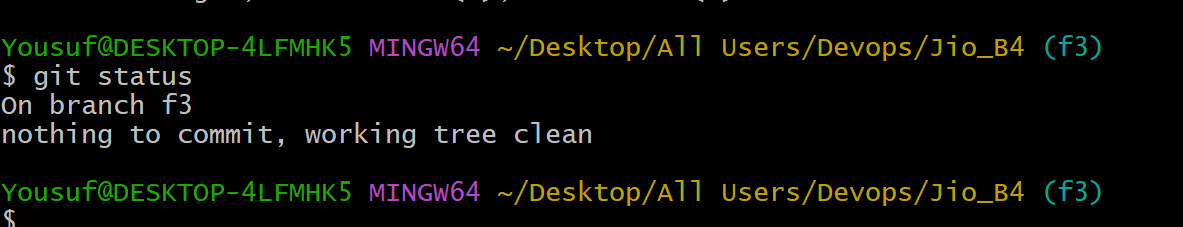


Execute the command ( git commit –am ‘version changed to v50’)



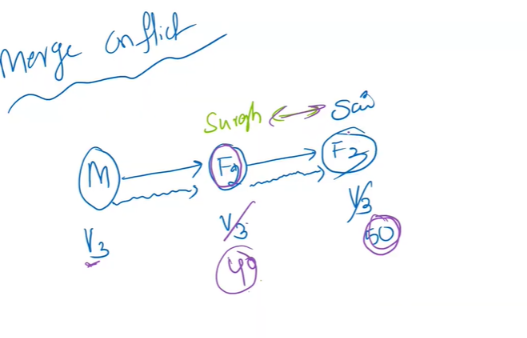


The code has been committed and the working tree is clean



1:12

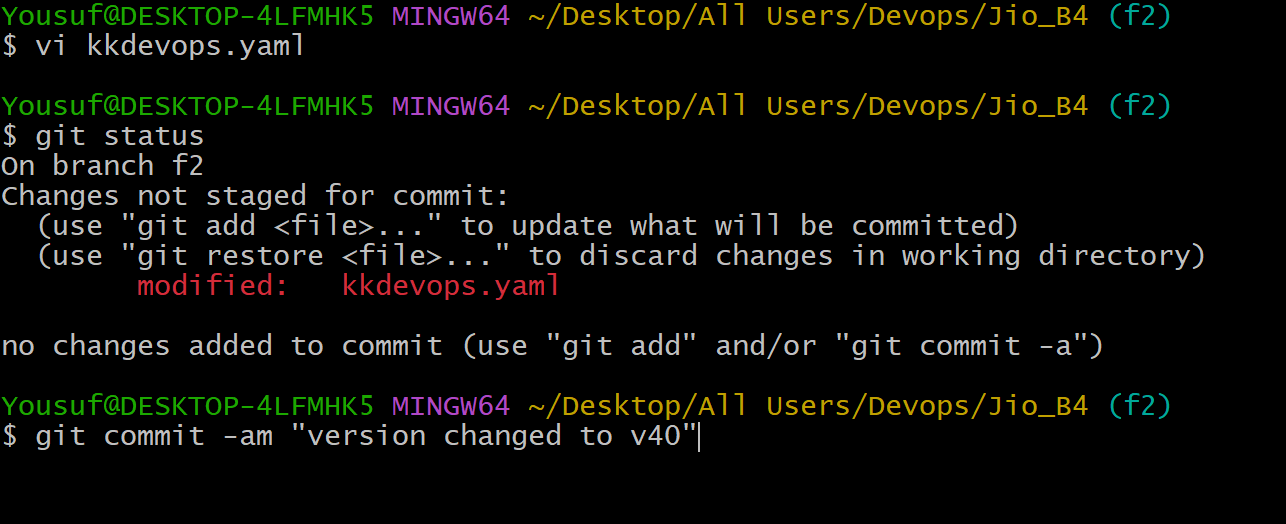
Let’s assume that suresh is working on F2 branch and he change the version to 40 in kkdevops.yaml and sai is working on F3 branch and he change the version to 50 in kkdevops.yaml



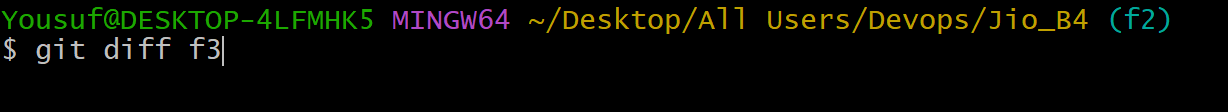
Let’s move the cursor to f2  
git checkout f2

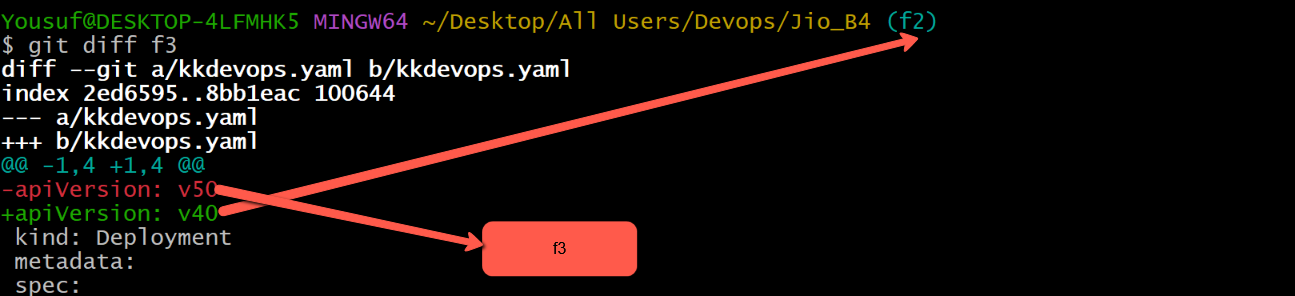
open the kkdevops.yaml file and change the version v3 to v40 and Execute the command ( git commit –am ‘version changed to v40’)



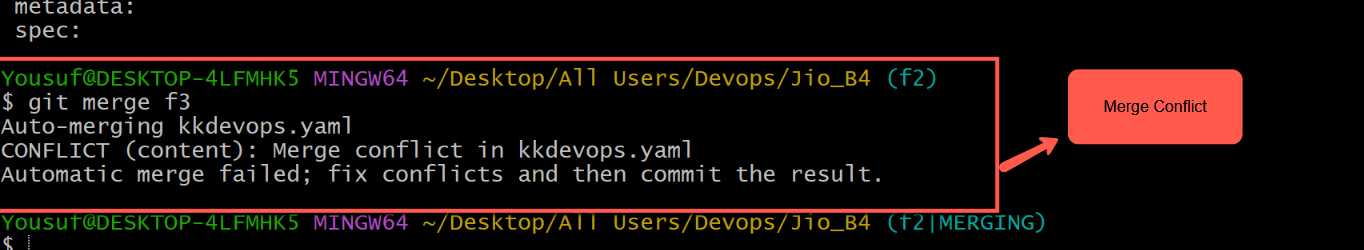


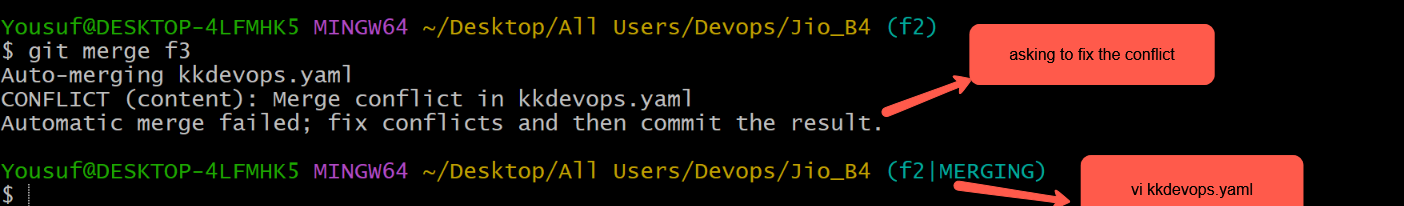
Now Here when we try to merge v50 of kkdevops.yaml of f3 branch witf v40 of kkdevops.yaml of f4 branch we will see the conflict message

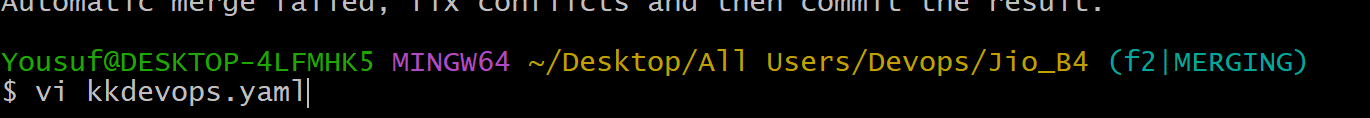


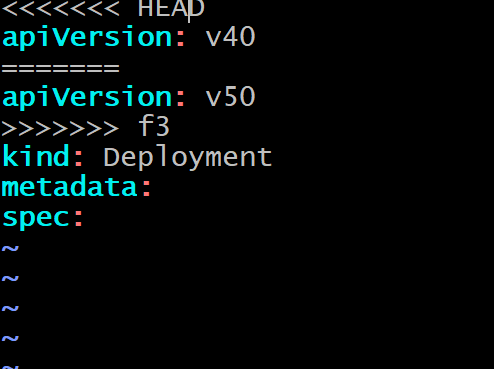


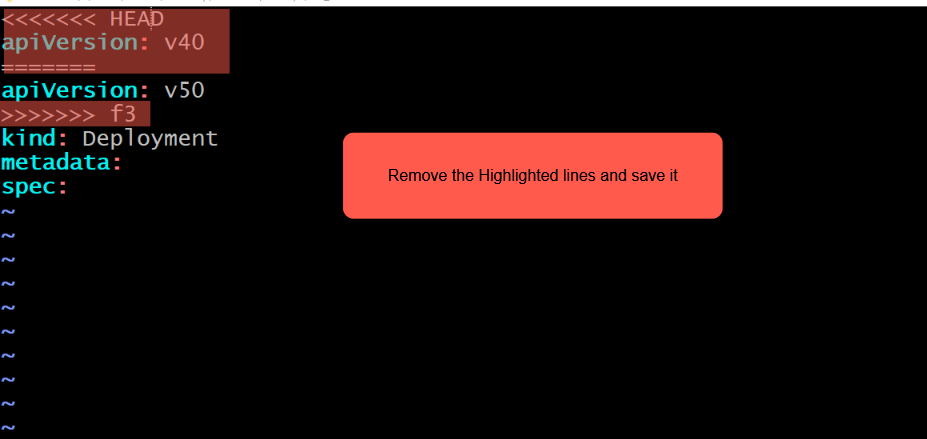
Now merge f3  

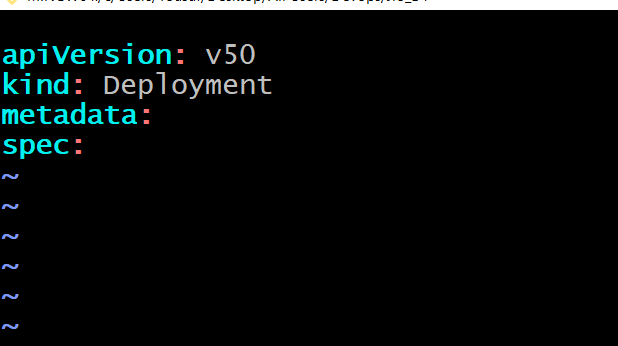







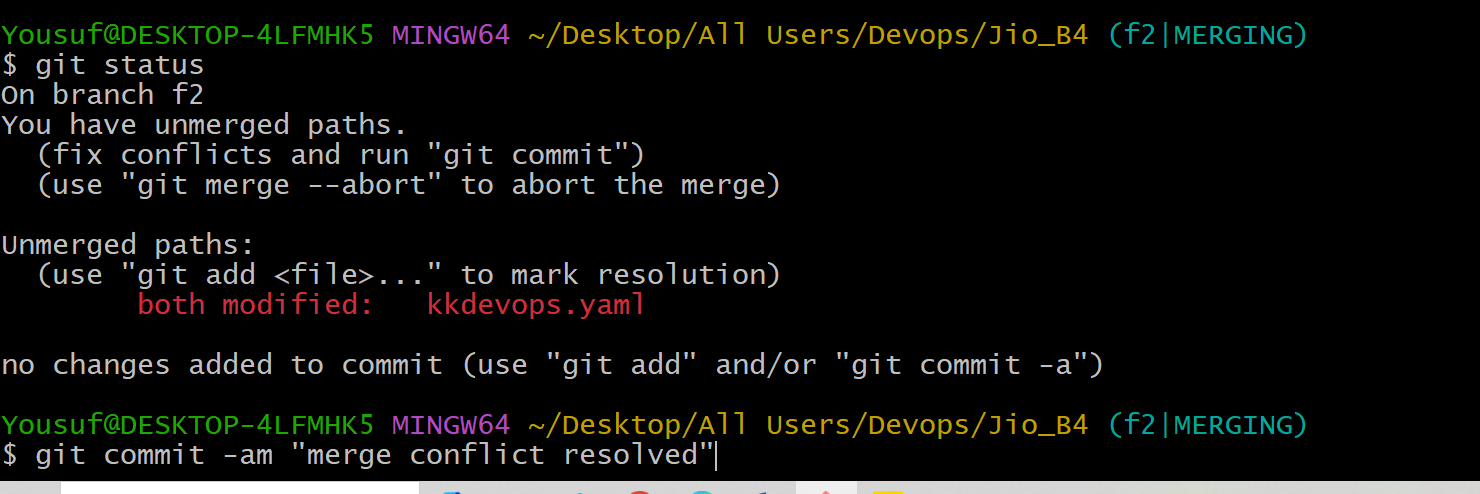


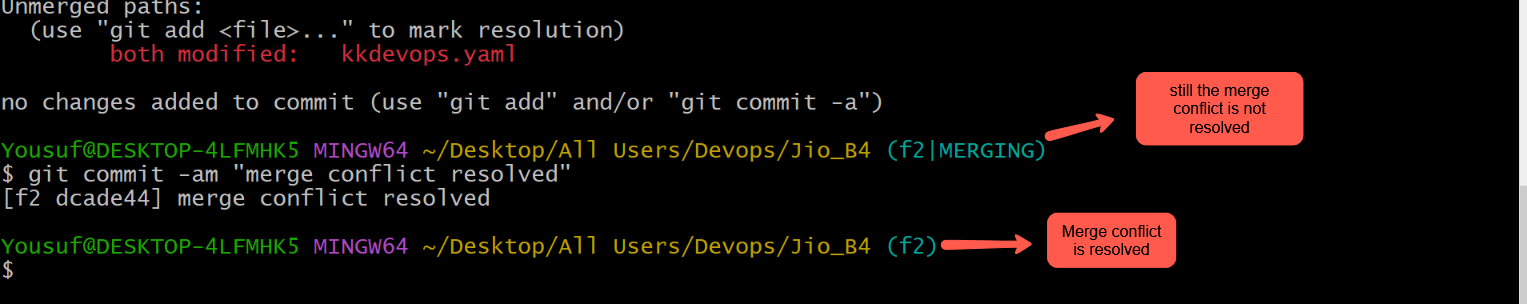
Discuss about this merge conflict with other person and come to conclusion to use only one specific version . here I am going with v50. So make the changes as shown in the below image  


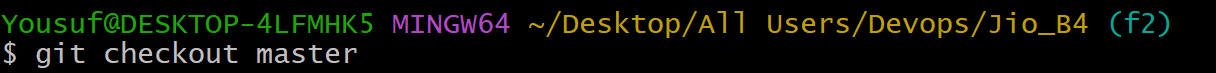


Save it   
but still there is merge conflict as shown in the below image to overcome this we have to execute the command git commit –am “”

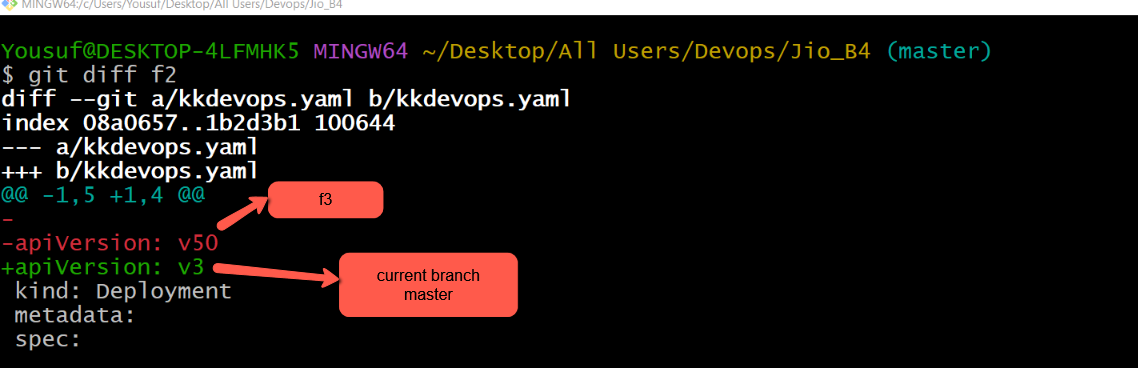
After executing the git commit –am “merge conflict resolved”  
The merge conflict will be resolved

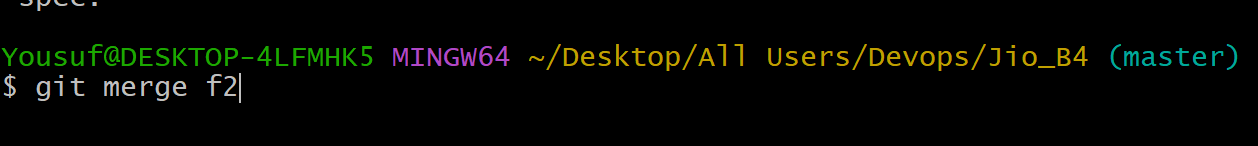


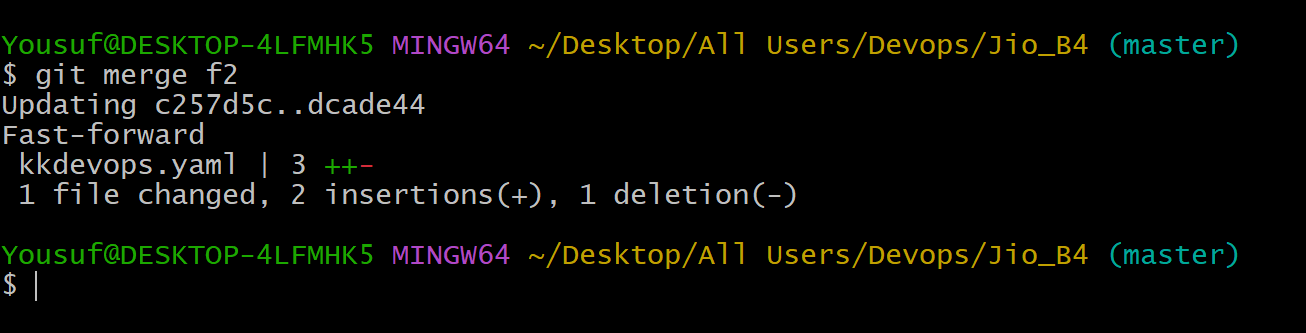


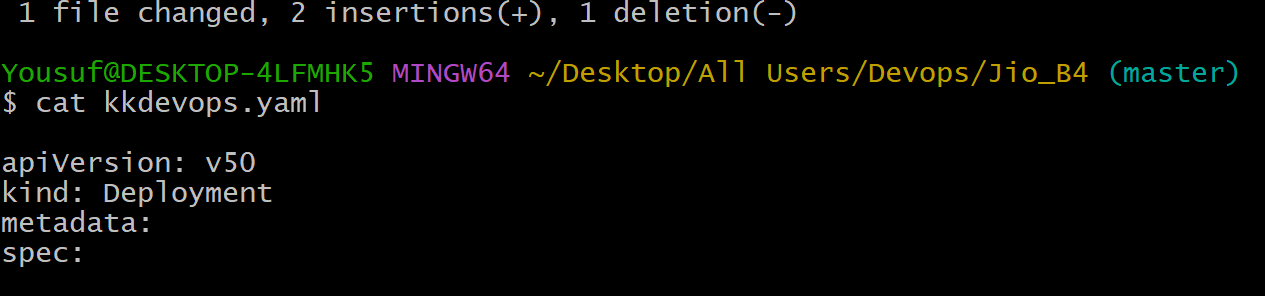
Now move to master branch   


Execute the command git diff f2  

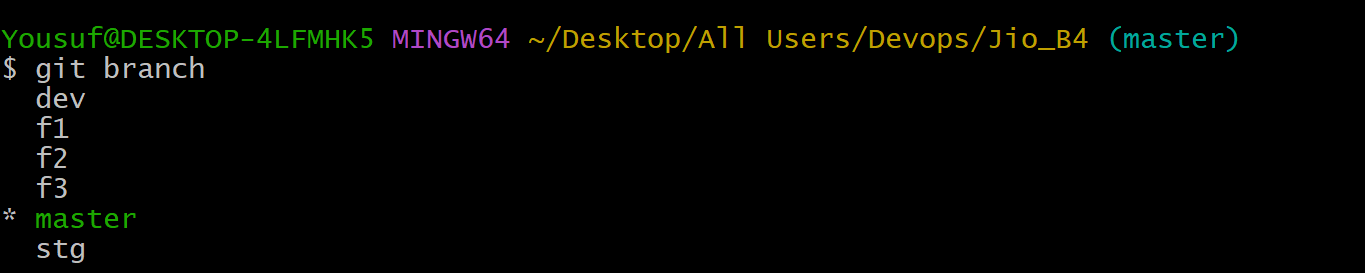
We have to merge f2 in master branch. So execute the command git merge f2  




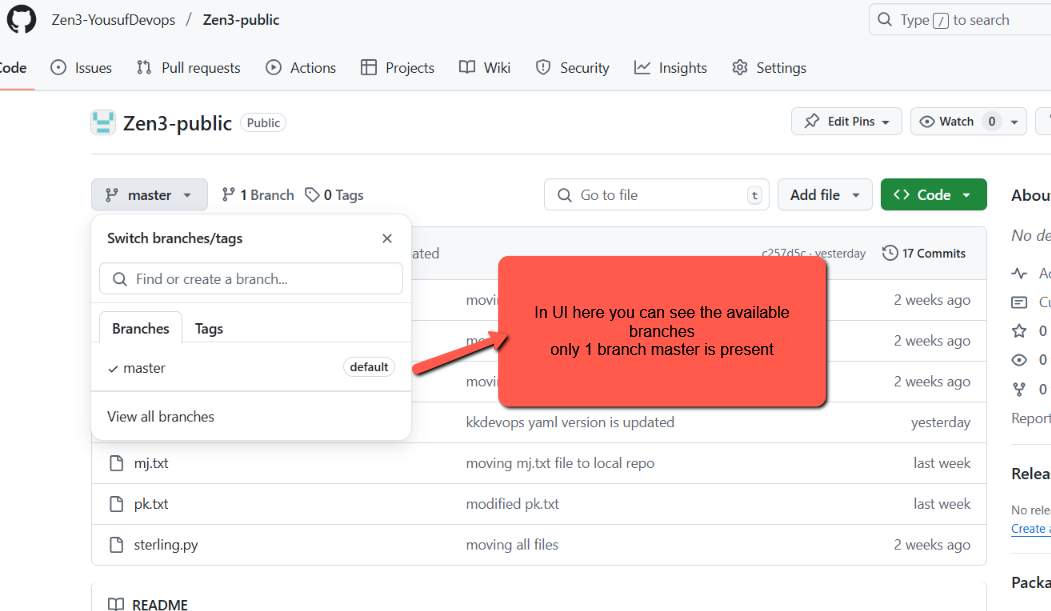
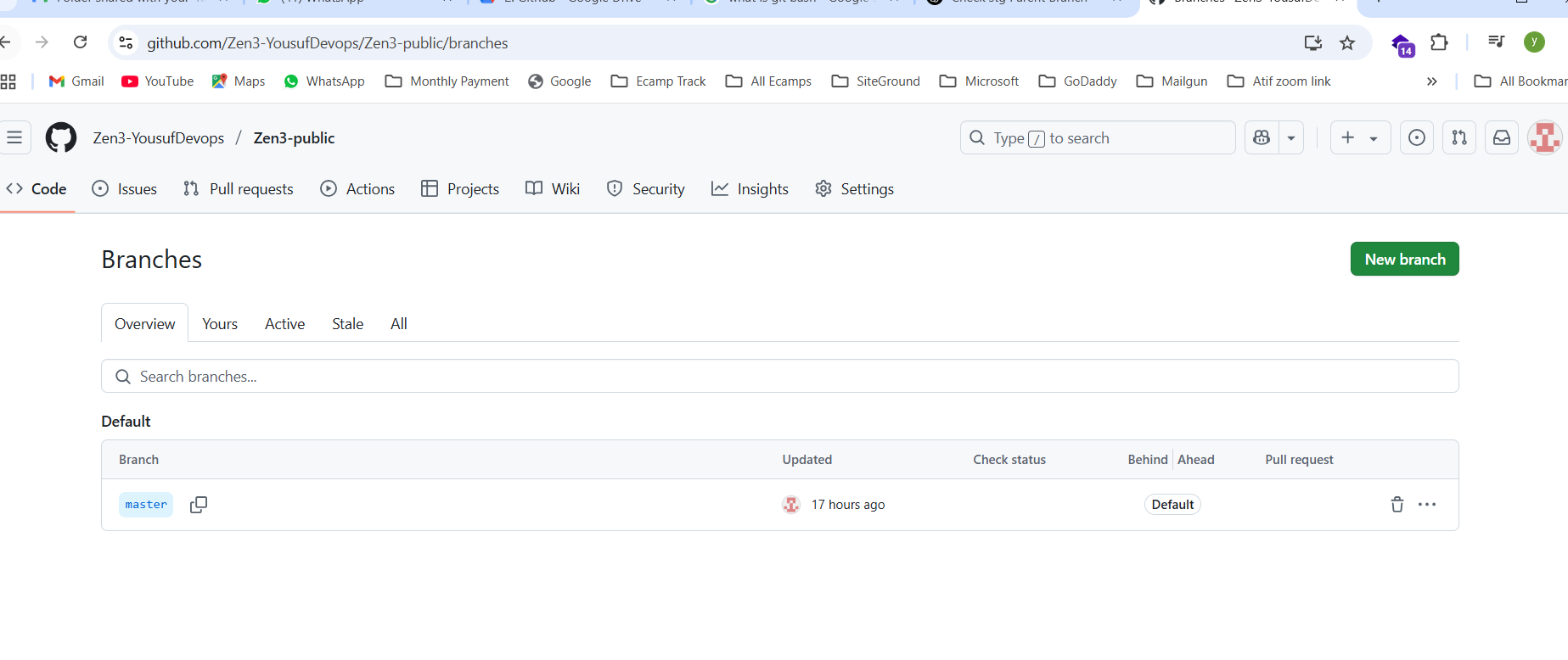
F2 is successfully merged 

ALL THE CODE IS UP TO DATE  
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

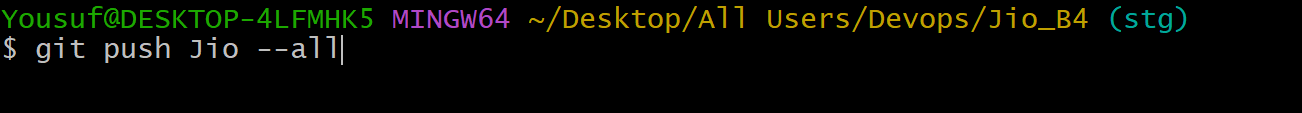
Now check the No of Available branches in Git Bash and in GithHub UI

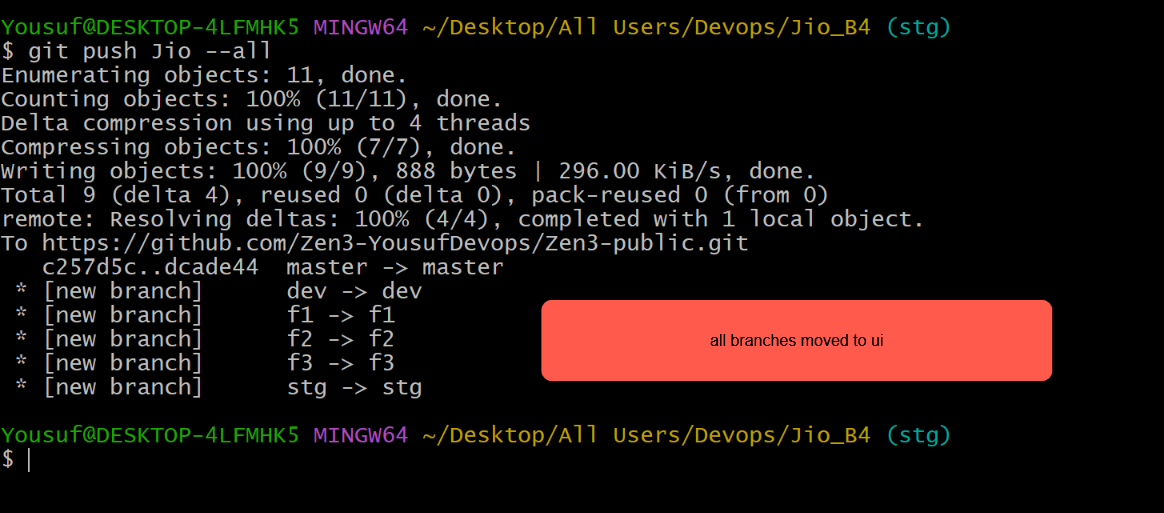


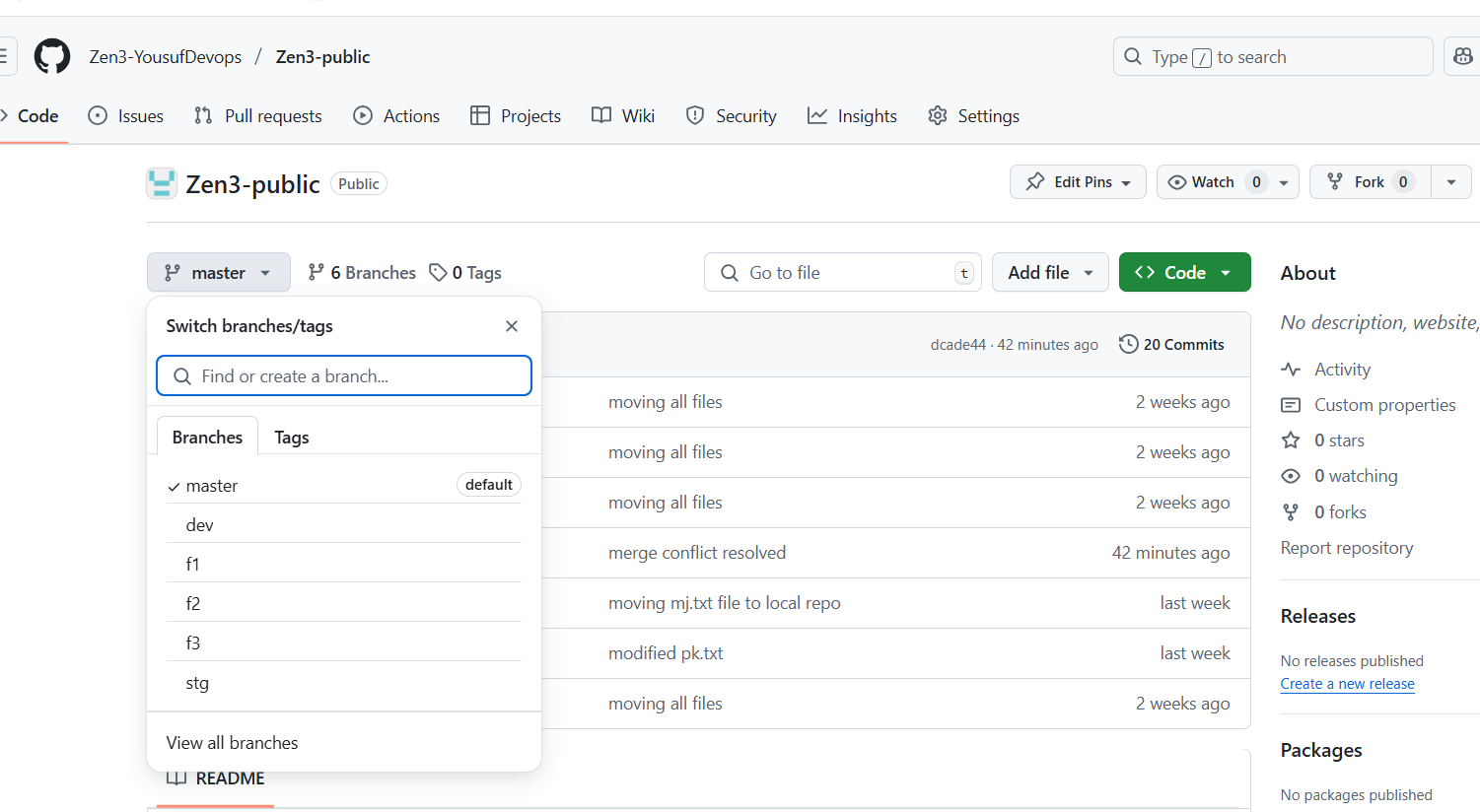
Only 1 branch (ie) Master branch is available in GitHub UI. Check the below image

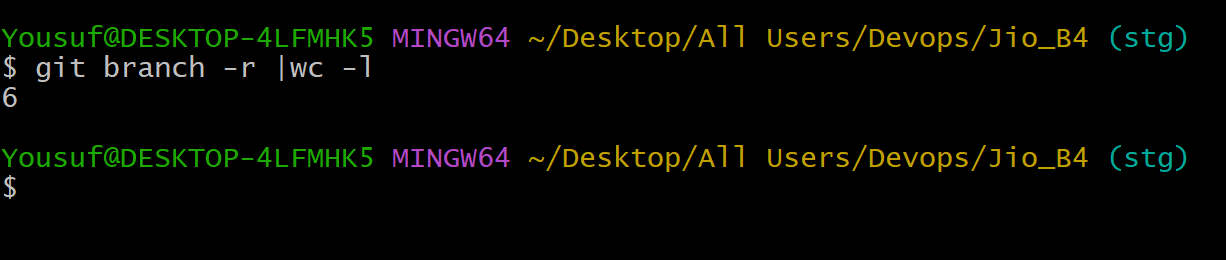
So to push all the local branches to github ui we have execute the following command git push Jio --all

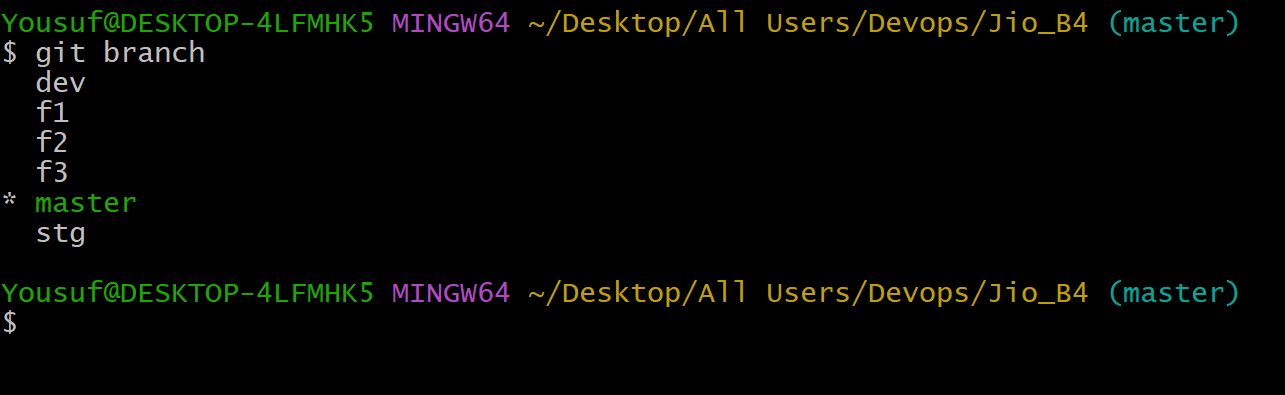
To push a particular branch to github ui we have to execute the command   
git push Jio master dev  
  


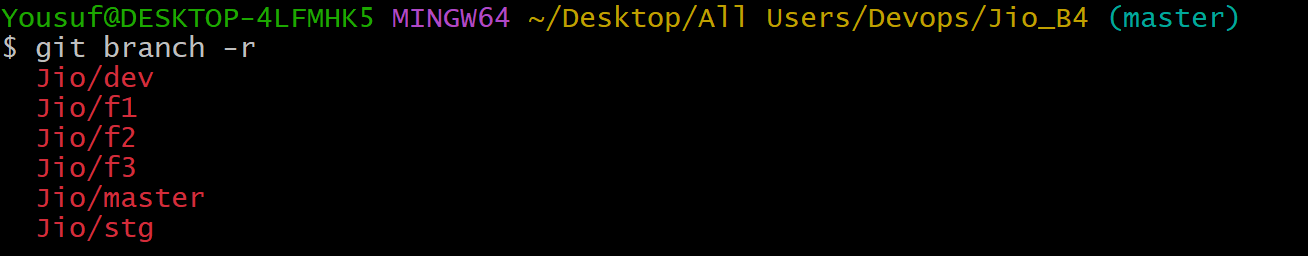


All branches are copied to GitHub UI 

To list the No of branches in remote Repository we have to execute the command

git branch –r | wc -l 

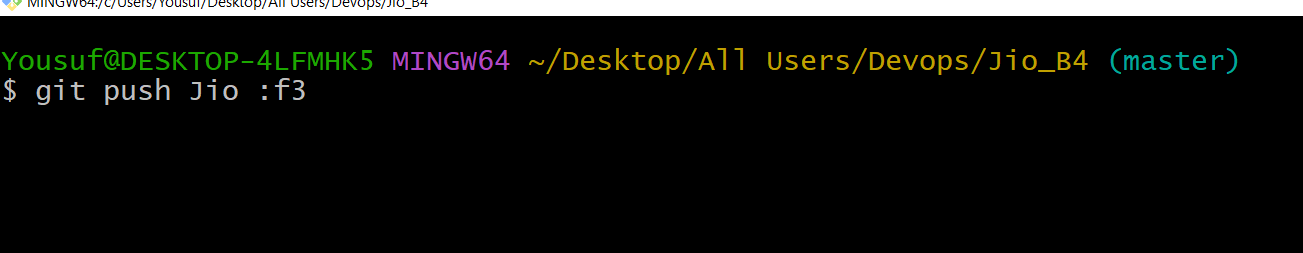
To list the branches present in Local Repository we have to execute the command  
git branch   


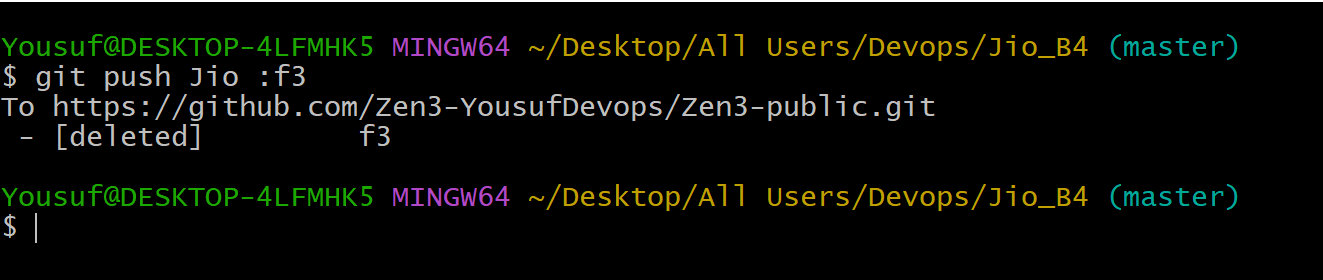
To List the branches present in Remote Repository we have to use the command git branch -r  


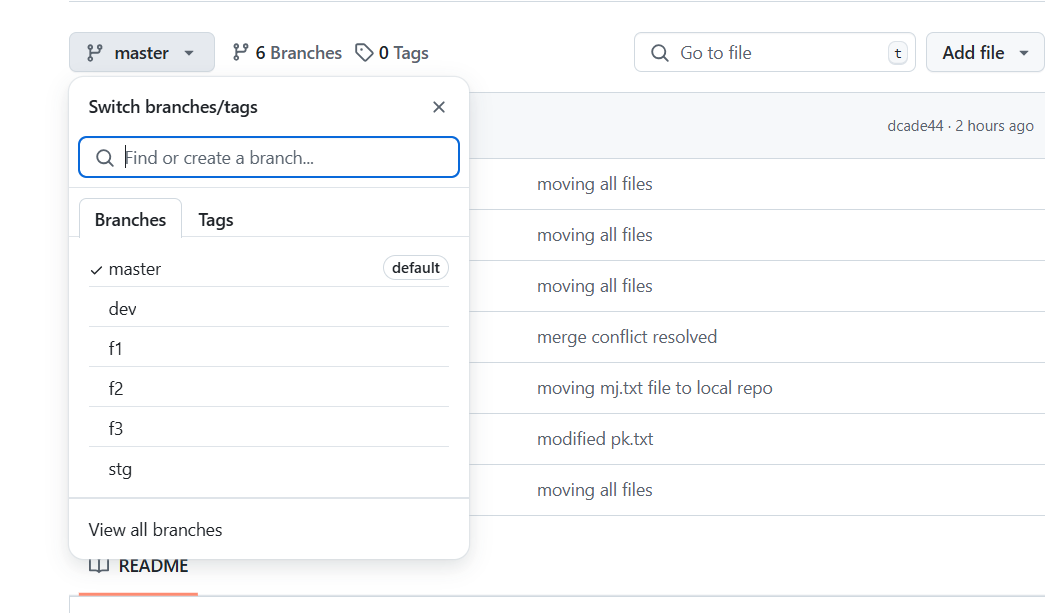
So to delete a branch in local repository we use the command ‘**git branch –d f3’ The branch f3 will get deleted but not get deleted in Remote Repository (ie) Github UI   
so to delete the branch f3 in remote repository we have to use the command**

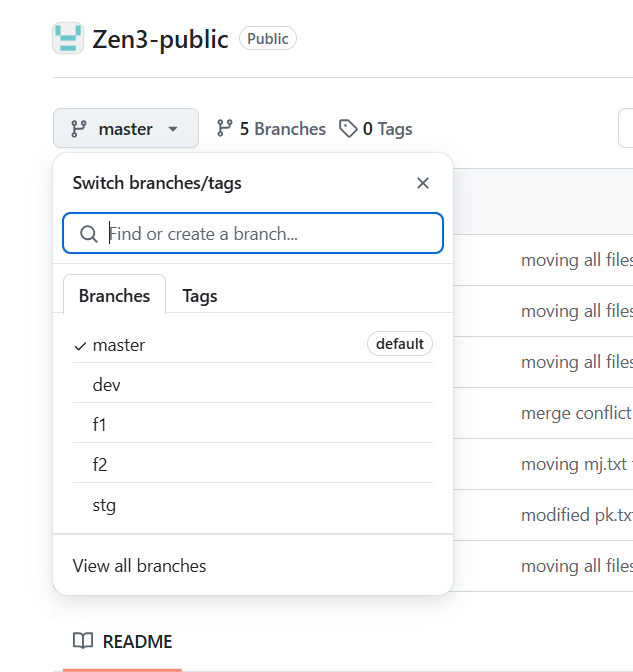
**“git push Jio : f3” (It’s a dangeours command)**

To delete the f3 branch in github rem **remote repository (github ui)we have to use “git push Jio : f3” **



Branch f3 in Remote Repository got deleted 

Before deleting f3 branch  


After deleting the f3 branch   


So the f3 branch got deleted in Github Ui  
  
End of the Session   
Technical question starts at 1:31:00