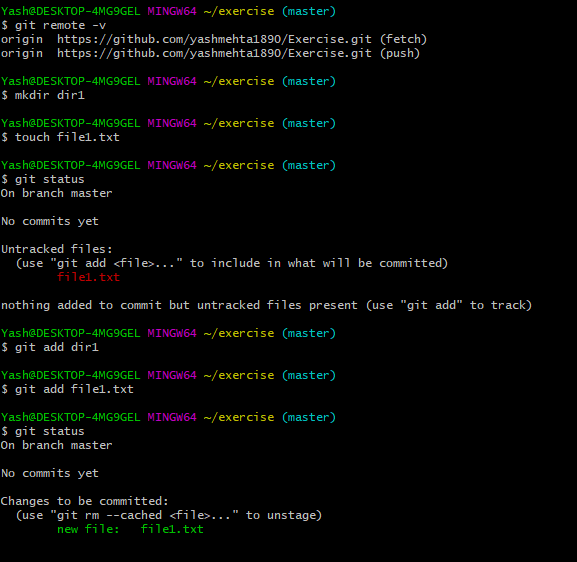
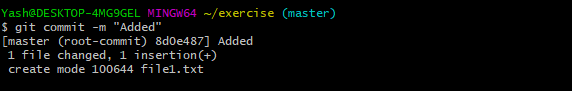
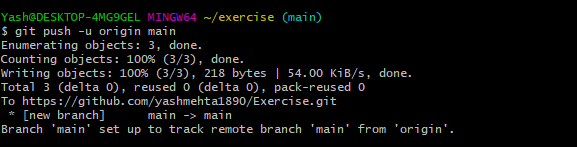
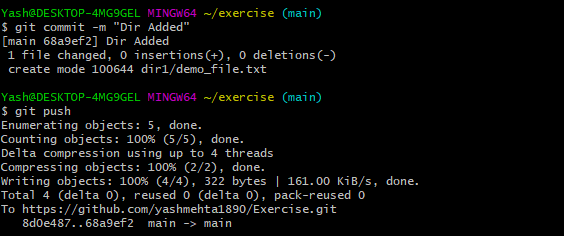
1. Create one repository in git, Clone it in your system, push directory and file structure in git.

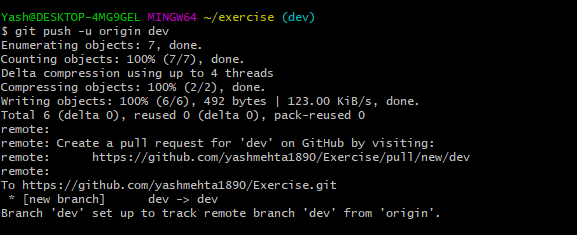
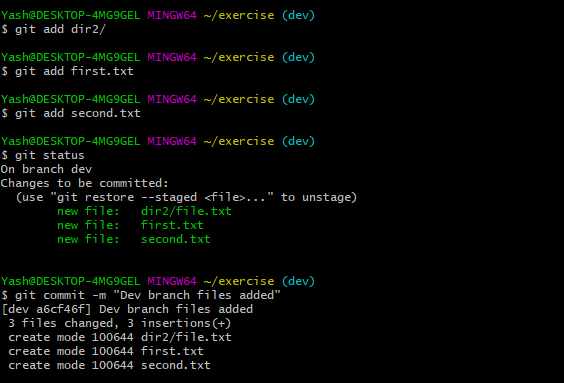
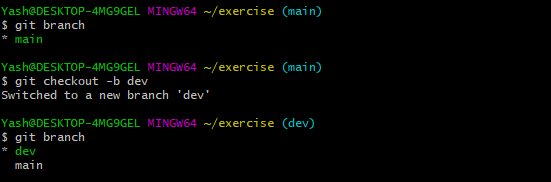




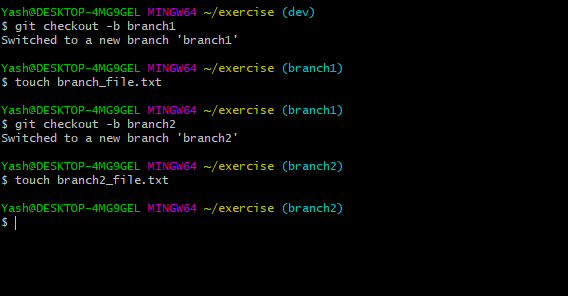


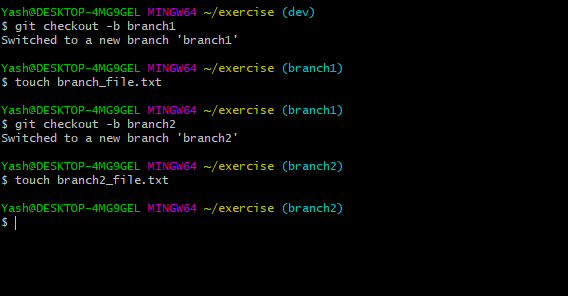


1. Create ‘dev’ branch from master and push some contents (directory with two files) inside this ‘dev’ branch.

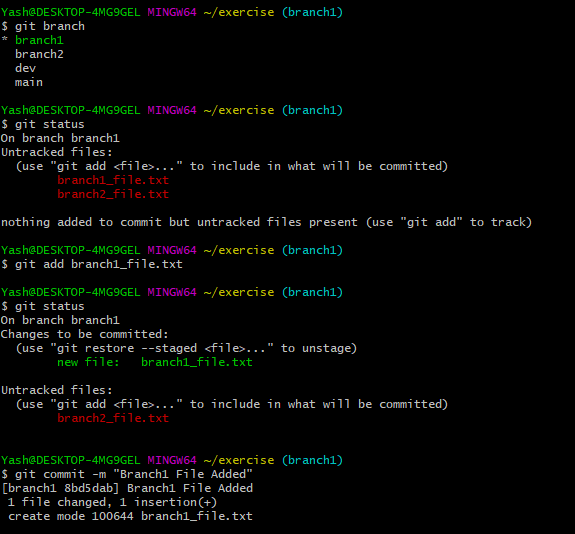


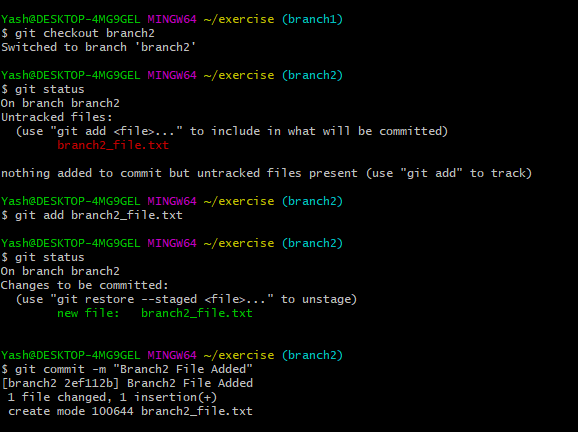
1. Create ‘branch1' branch from ‘dev’ branch, create one file in ‘branch1’, as well as create ‘branch2’ branch and add one file for this branch too.



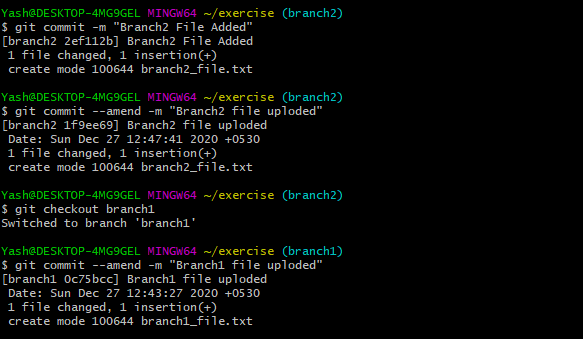


1. Change some contents in existing file inside ‘branch1’ branch, and do same thing for ‘branch2’

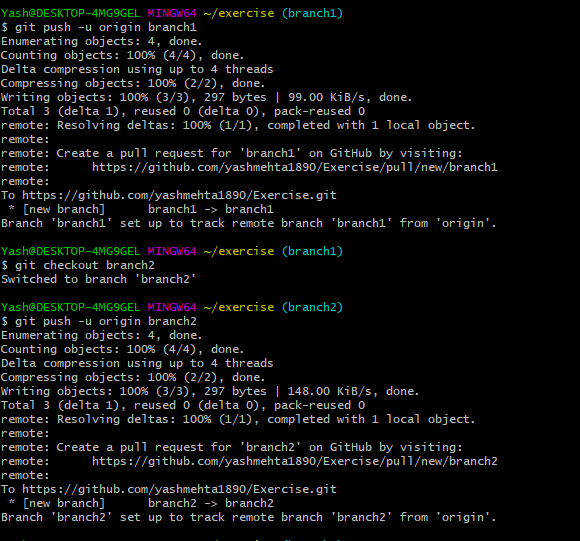




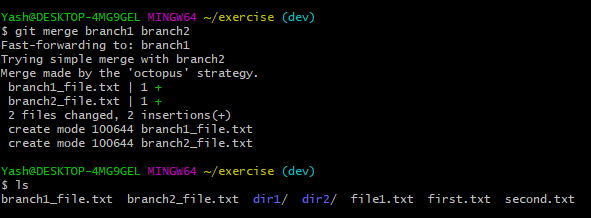
1. Commit those changes and after committing, change the commit message.



1. After changing commit message you have to push your new changes on git. (NOTE: if you get any conflicts then resolve it and then push that in git.)

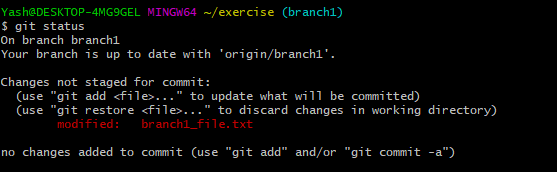


1. Merge ‘branch1’ and ‘branch2’ in dev branch.

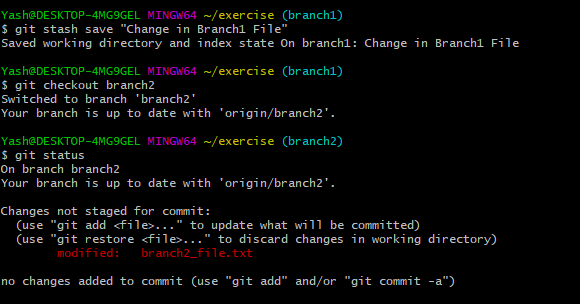


8. Follow the steps:

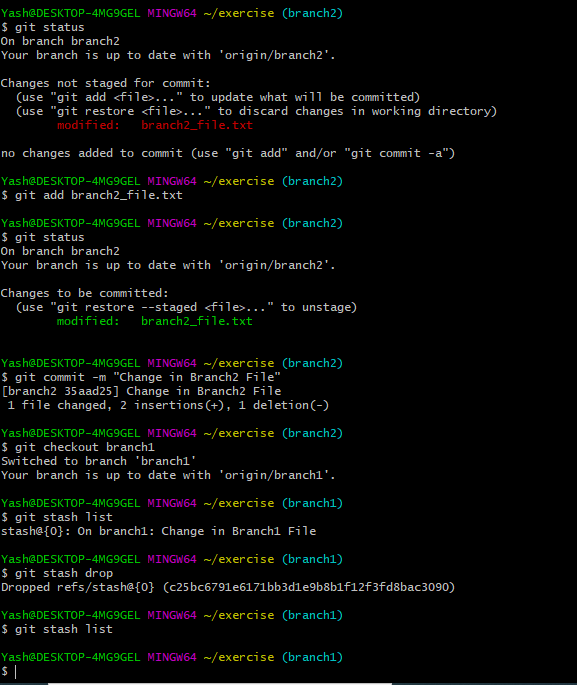
(a). Make a few changes in ‘branch1’ contents



(b). Without ‘adding and committing those changes' (branch1), make some changes in branch2 contents (stash your branch1 changes and then apply some changes in branch2)



(c). Add and commit your branch2 changes and then come back to branch1 and get back your stash changes.

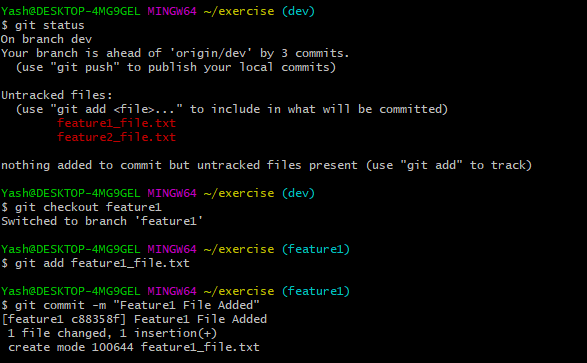


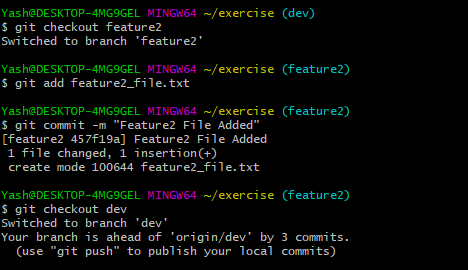
9. Follow the steps:

(a). Create two branches from dev, i.e ‘feature1’ & ‘feature2’.

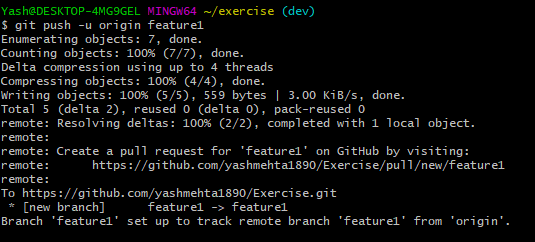


(b). Add some content in ‘feature1’ as well as add some content in ‘feature2’

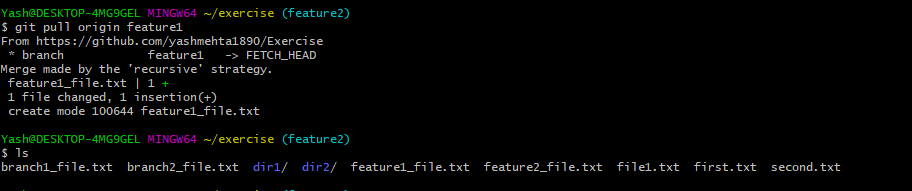




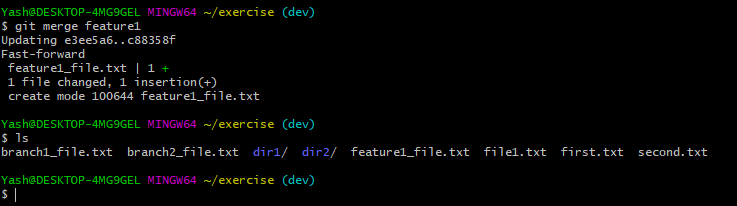
(c). Push ‘feature1’ on git



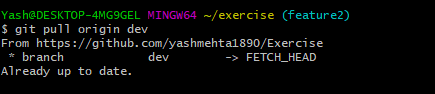
(d). Take pull of ‘feature1’ in ‘feature2’



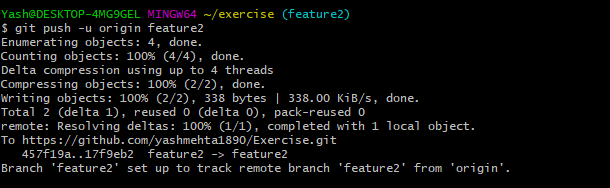
(e). Merge ‘feature1’ into ‘dev’ branch



(f). Take pull of ‘dev’ branch



(g). Push your ‘feature2’ on git



(h). Rebase ‘feature2’ with ‘dev’ branch and then merge ‘feature2’ in ‘dev’.

