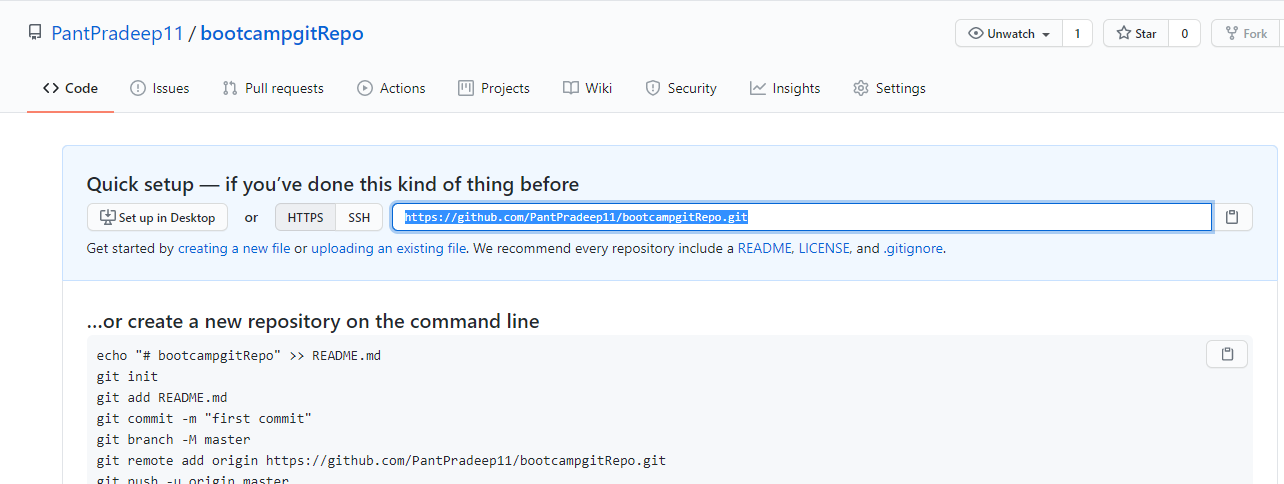
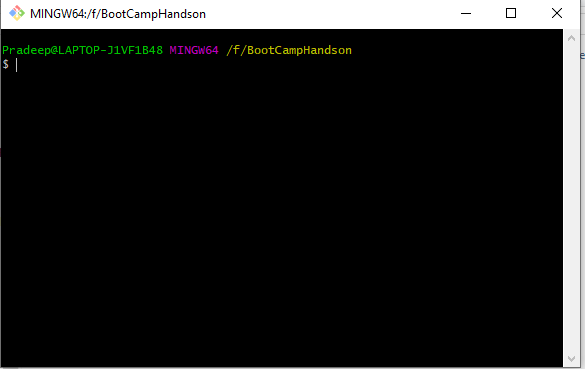
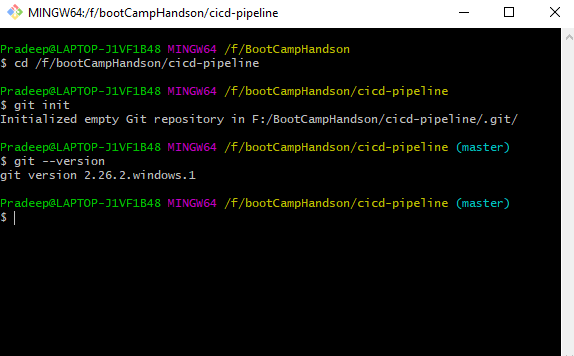
1. Created a new repo in git



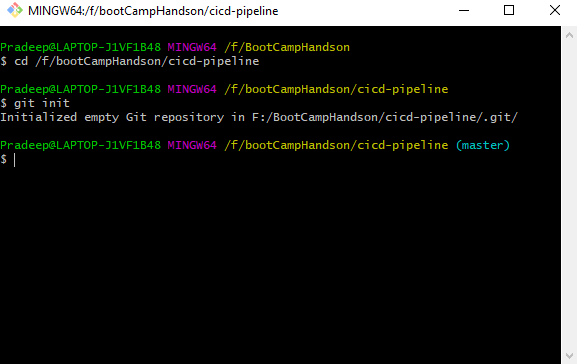
1. Open git client on your machine and move to the code path



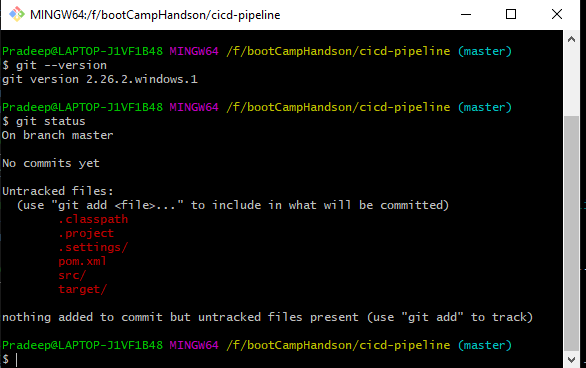
1. Check git version.



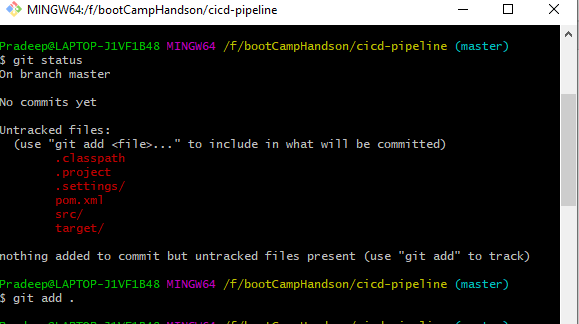
1. Initialise the git.



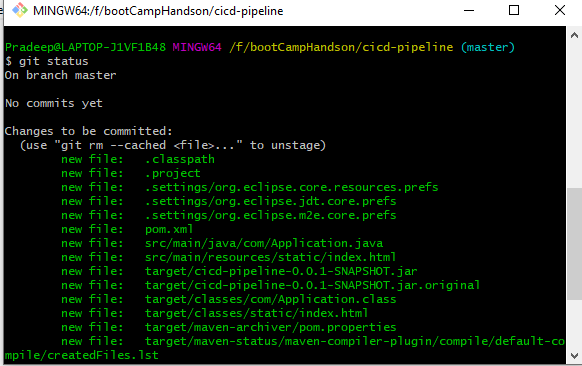
1. Check the untracked file using the git status



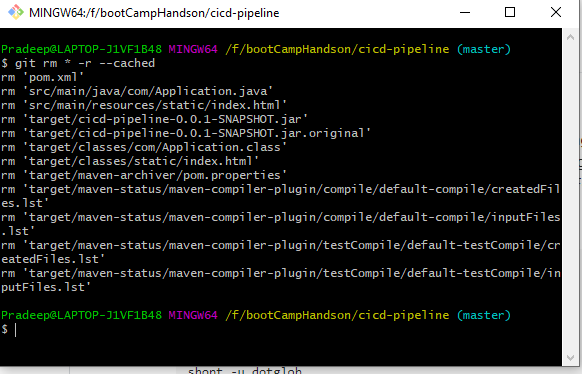
1. Add the files in staging area by using git add <filename> for individual or git add . for all files



1. Check the status of the staging area using git status.

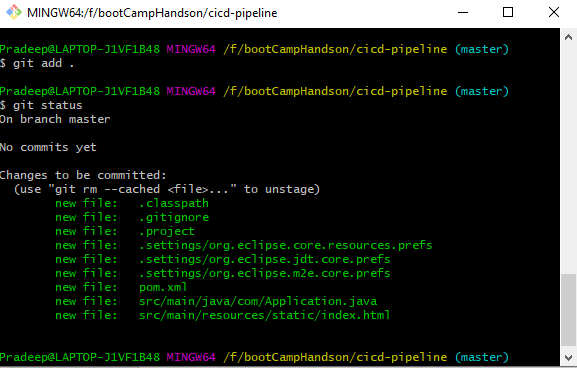


1. Now we can see many files of target folder also included so we can remove all the files using git rm –cached from staging area command then clean our mvn project to clearout the target folder. Or we have on more way to handle this using .gitignore file.

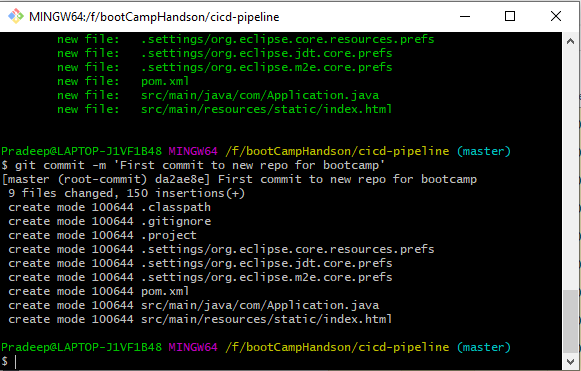




1. Added again all files using git add now the content from target folder is not included.

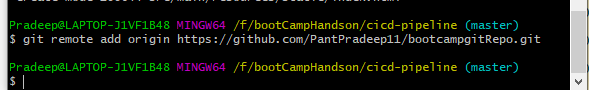


1. Commit the all staging changes to local repo.

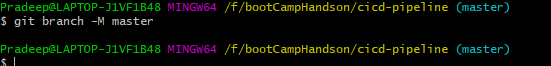


1. Linking the remote repo with your local repo using

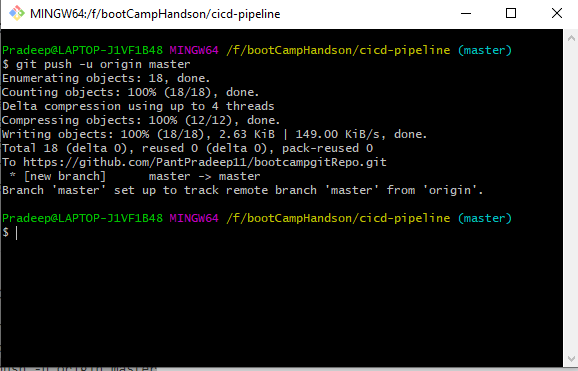
git remote add origin <https://github.com/PantPradeep11/bootcampgitRepo.git>



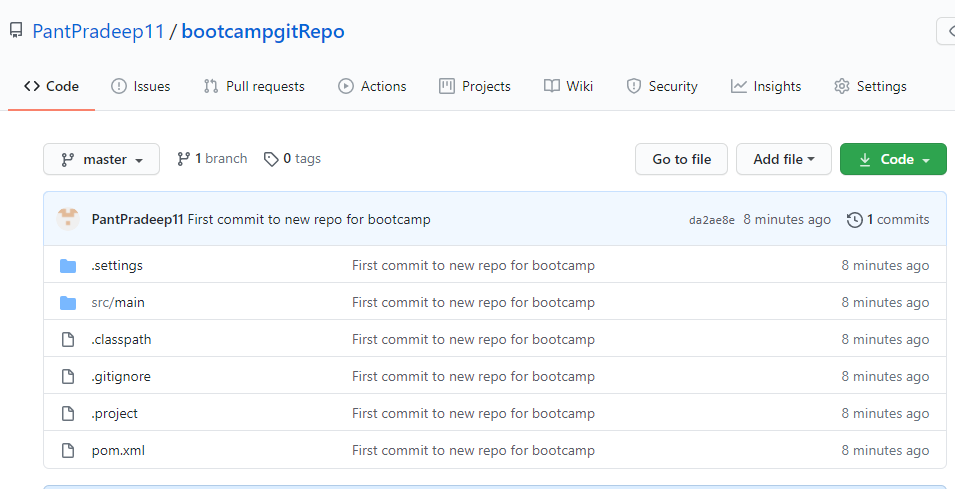
Map it with master branch.



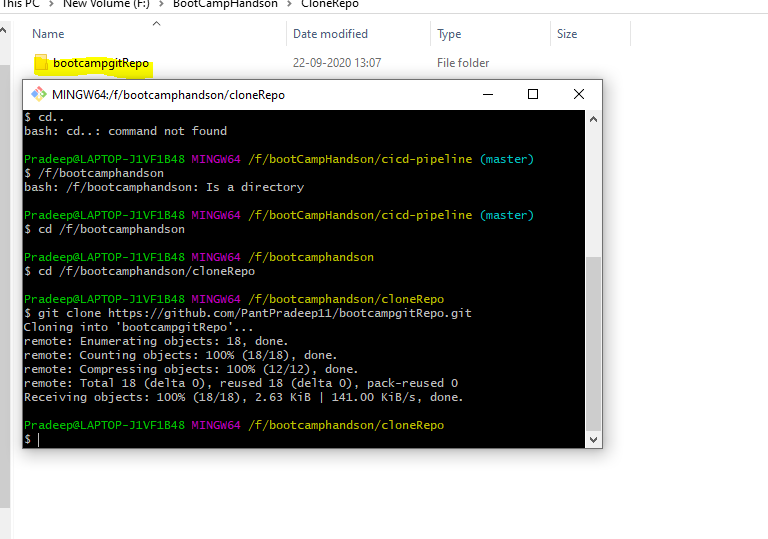
Push the changes to the remote repo.



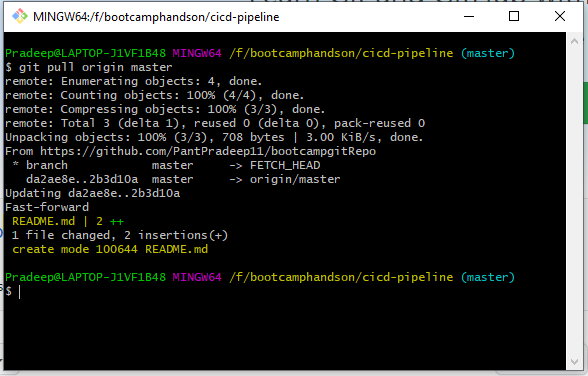
1. All files are pushed to repo and available at github.com.



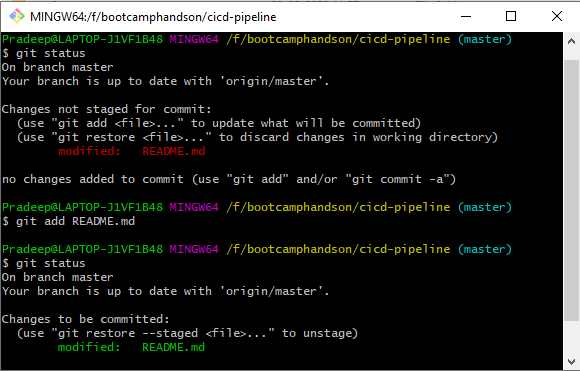
1. Cloning a git repo using git clone.

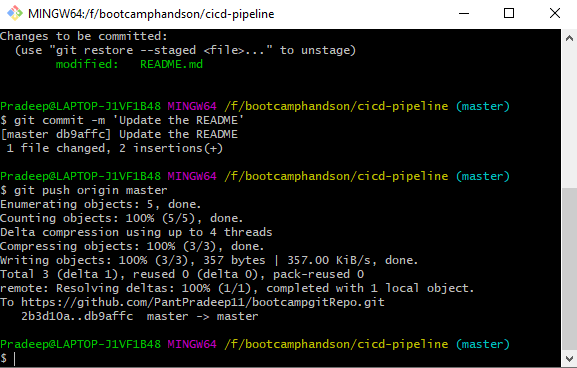


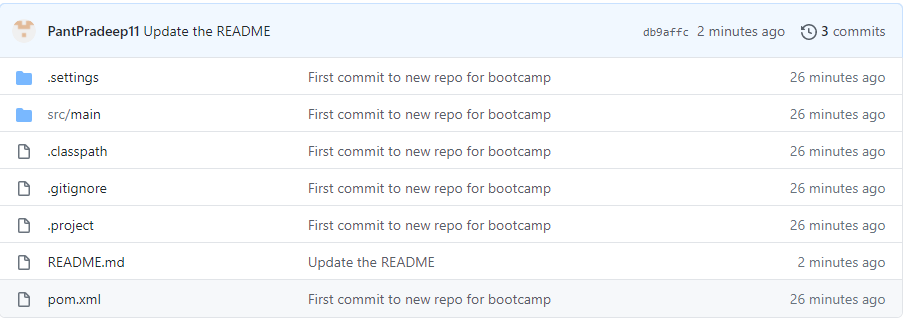
1. To check this, I have added on README file in github.com then use git pull to get the latest of remote repo.



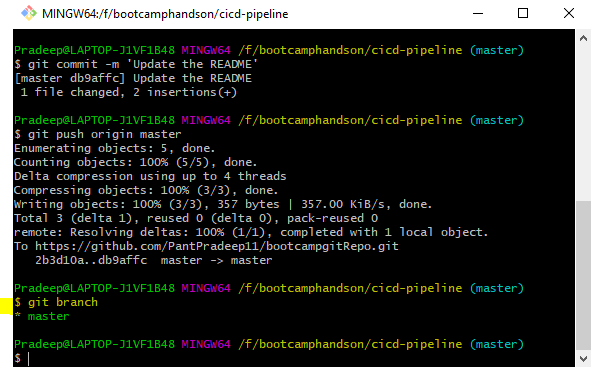
1. Updated that README at the working copy then push back to Remote repo.



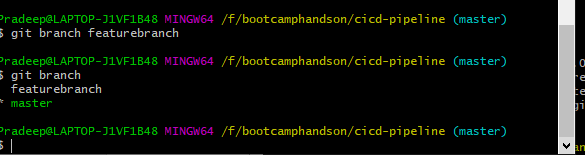




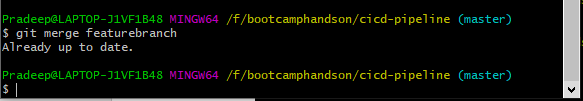
1. List all branch



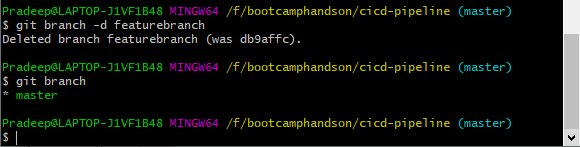
Create a new feature branch.



Merging the branch.

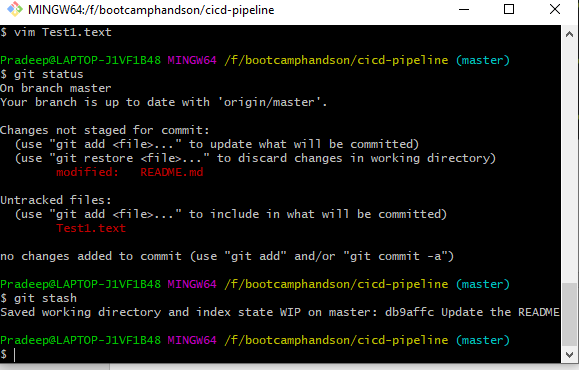


Deleting a branch

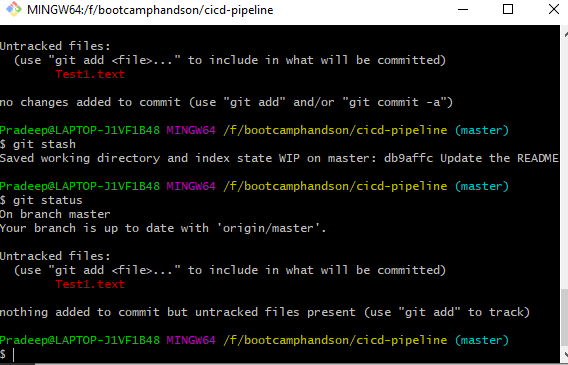


1. Advance Git commands.

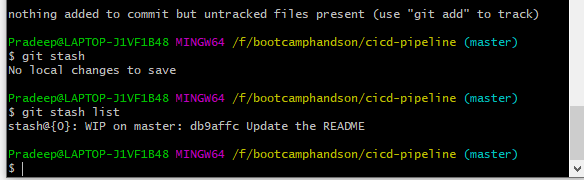
Git stash: added a new file and updated the existing file, then run git stash command



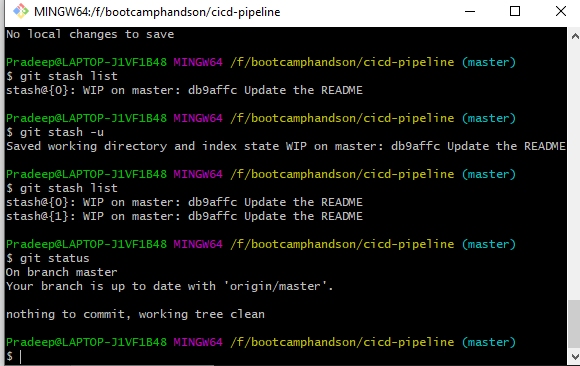
After stash command the modified file is stashed, but newly added file is still in untracked files we will use git stash -u for that.



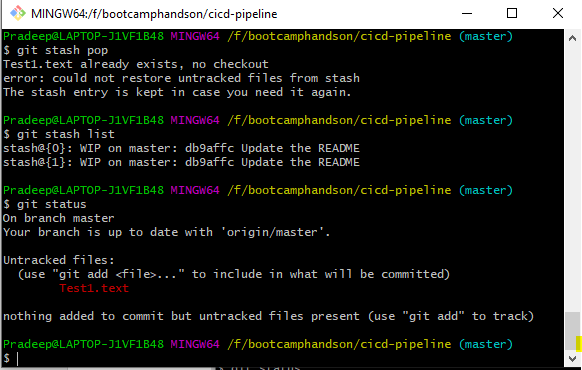
View all the Stashed changes



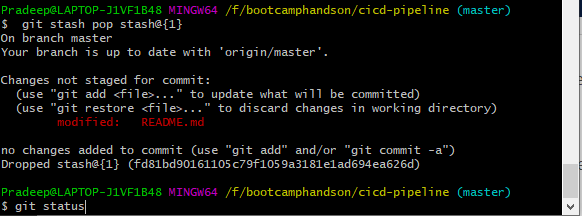
Git stash -u for untracked files



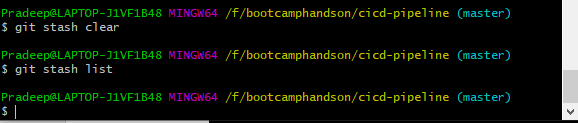
Reapplying the stashed changes



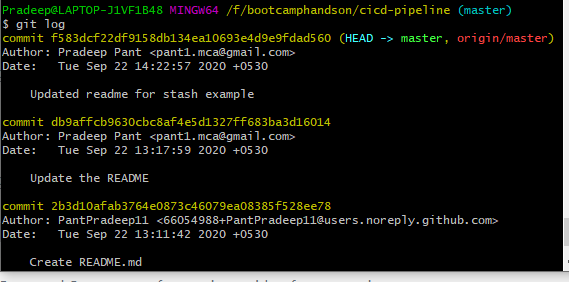
We can get specific stash by



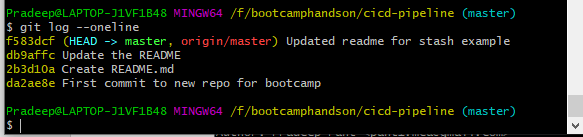
Cleaning up stash



Git log command:

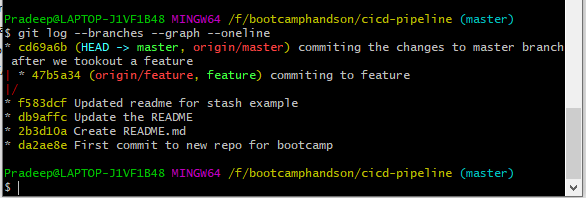


To display only summery of git commits

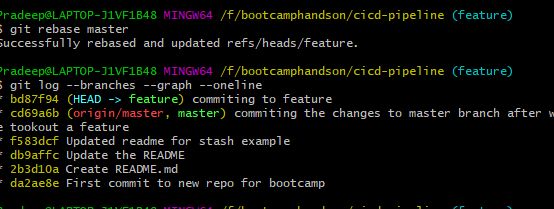


Git Rebase command:

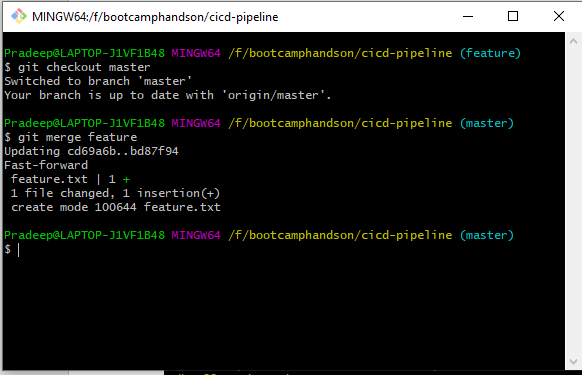
Added a new feature branch then make some changes in it and committed the same. Then again add something in master branch.



Git rebase command updated the master at feature.



Merging the feature branch changes in master after rebase.



Git revert command

