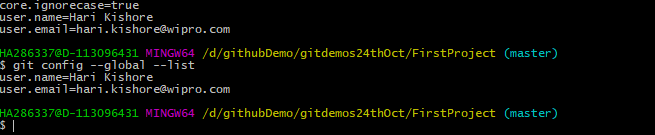
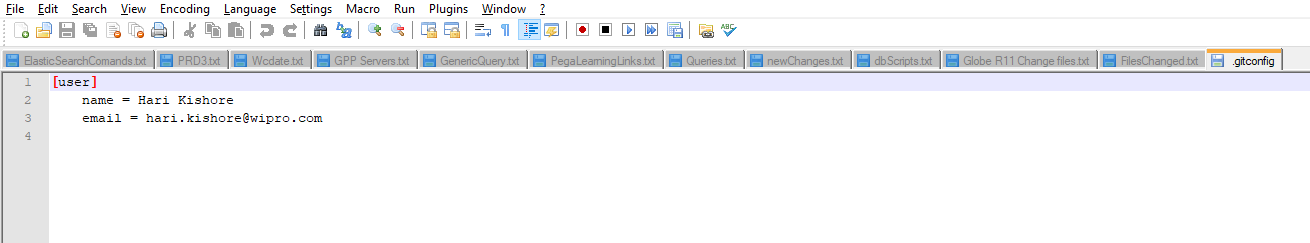
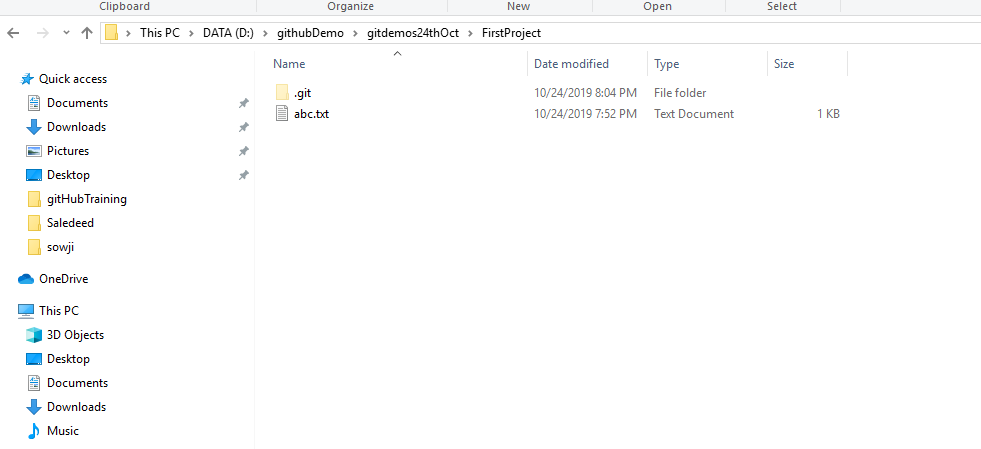
While doing the below labs, you need to take screenshot for each step, for each assignment and create word document. Upload the word document to topgear.

1. Set the global configuration file with your user name, email and editor as Notepad++. List all the properties which you just set.

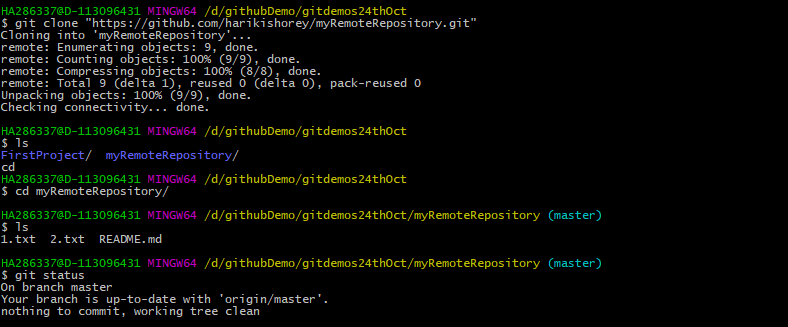


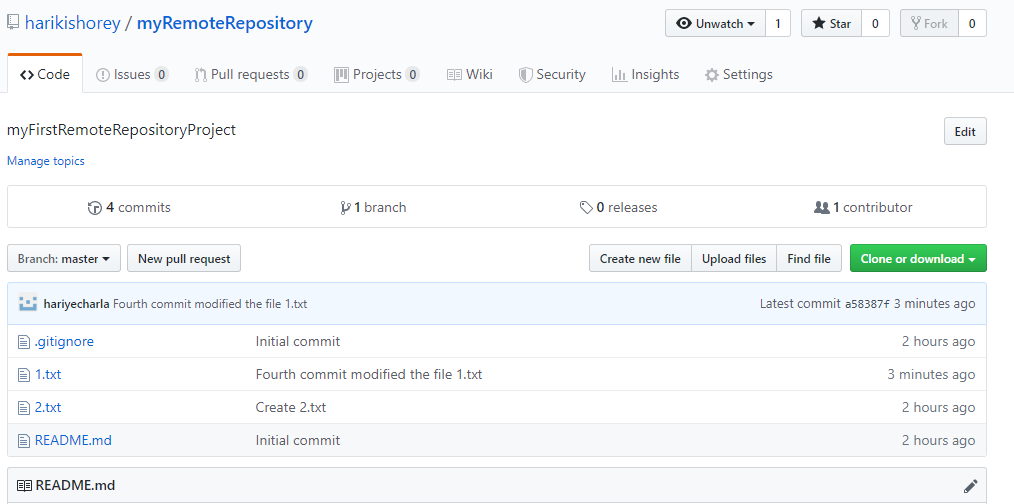


1. Make a fresh Git project

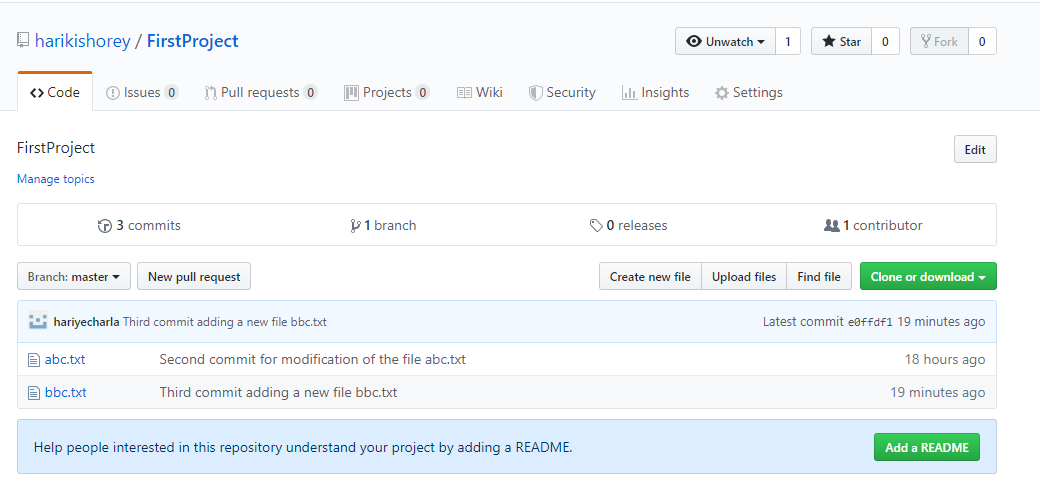


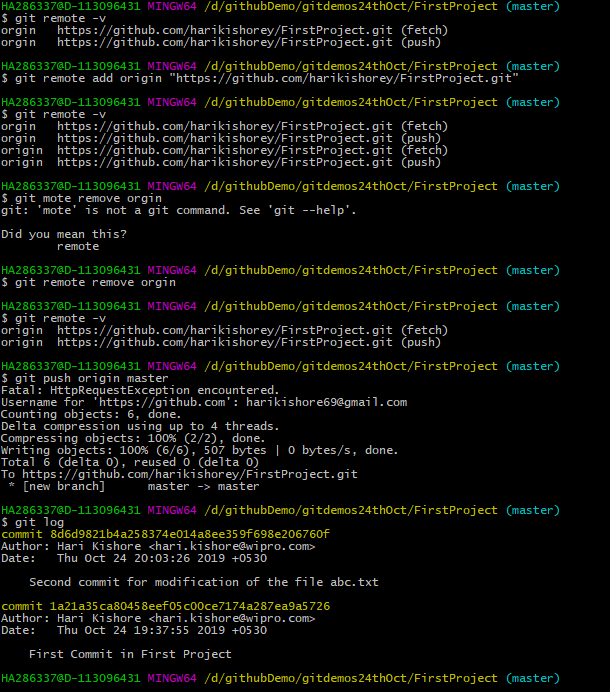
1. Create a Github account (Or use the account if already registered). Clone a project from the remote repository to your local repository



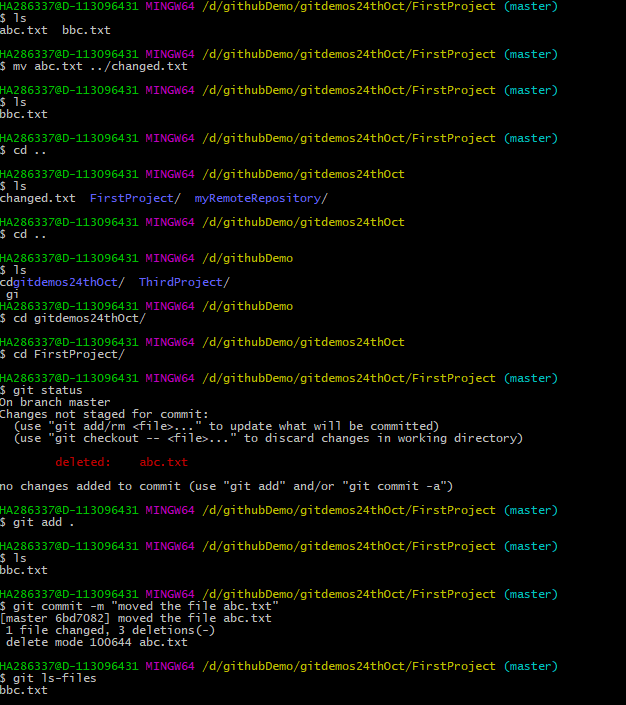


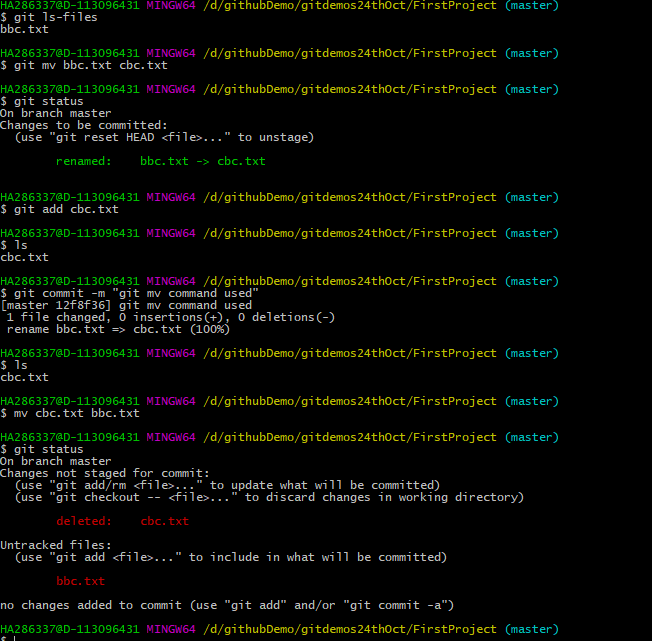
1. Push the project created in assignment 2 to the remote repository.



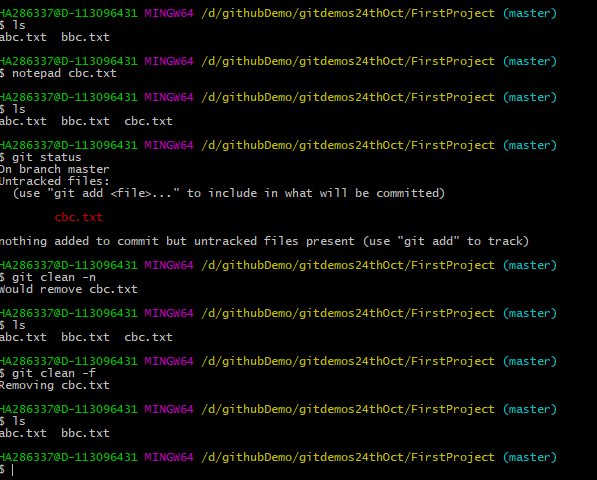


1. Use the different ways of renaming and moving files

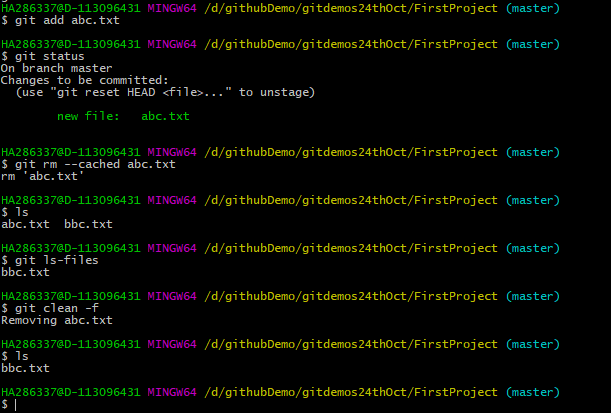




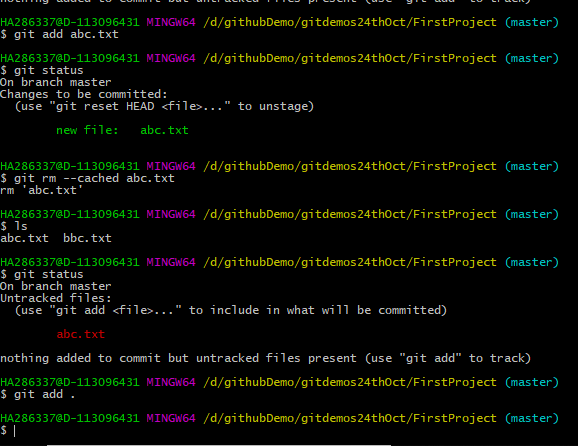
1. You just created a new file, but then you decided that the file is to be removed. How do you delete this untracked file.



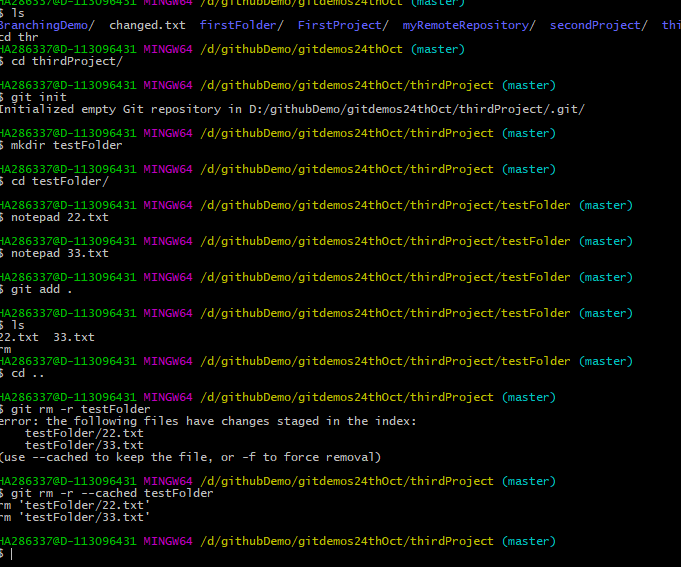
1. Demonstrate the following:
   1. delete of a tracked file



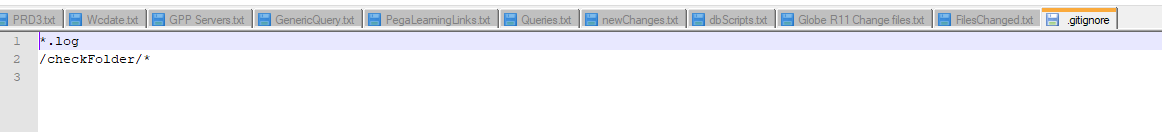
* 1. backing out staged deletion



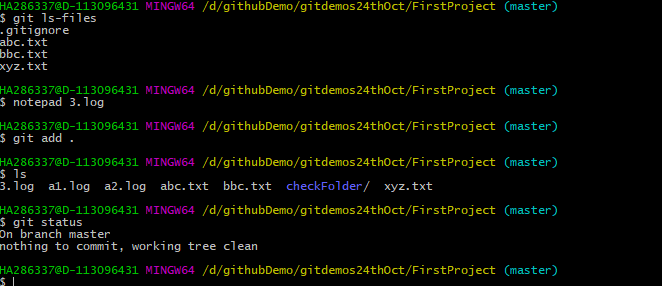
* 1. recursive deletion



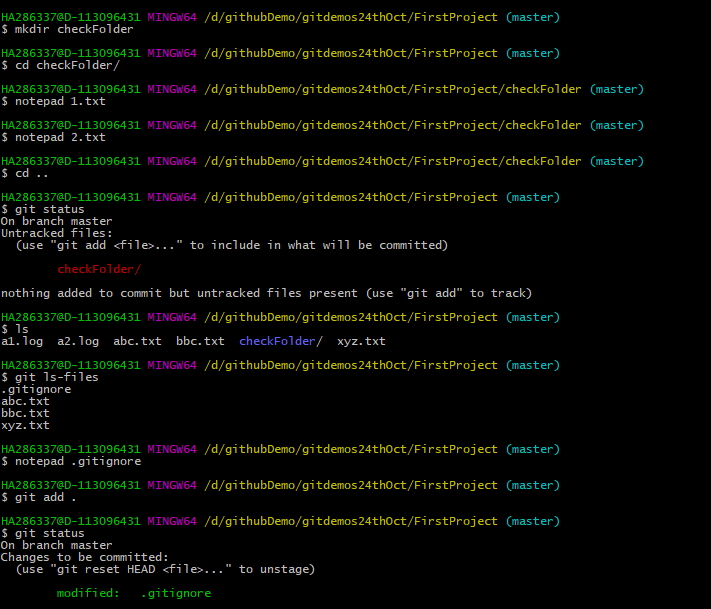
1. You do not want to push certain folders/files of your project; how do you manage this?



Trying to add 3.log file but because of .ignore file \*.log files not even showing the staged state



I am have given the property in .ignore to avoid the folder checkFolder. The below screenshot showing that



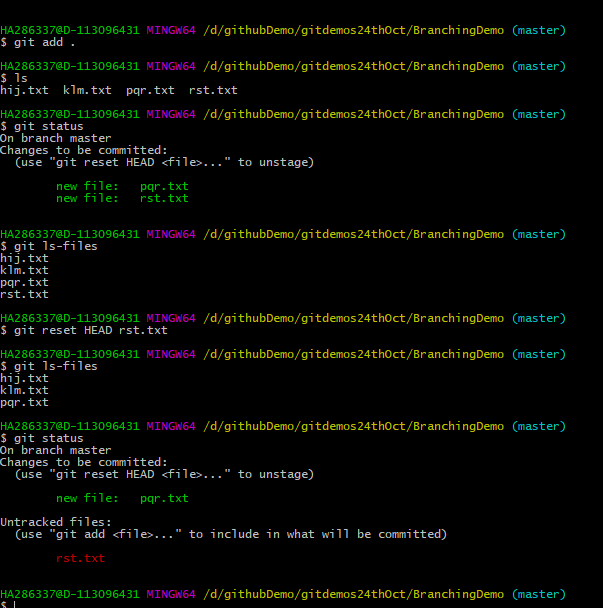
1. Create a branch called “test”. Make some changes in the master branch. Let there be some changes in the working directory and some in the staging area. Make some changes in the test branch as well. Issue the command to show the differences for
   1. Working directory vs Staging area

Working directory: where all the physical files residing

Staging area: Any new file created it is located in the working directory and the git status is showing as untracked file. So we need to stage the file using git add <file Name>. If we do any modification also we need to do the staging in this case the file is tracked because we already ran the command git add for the specific file.

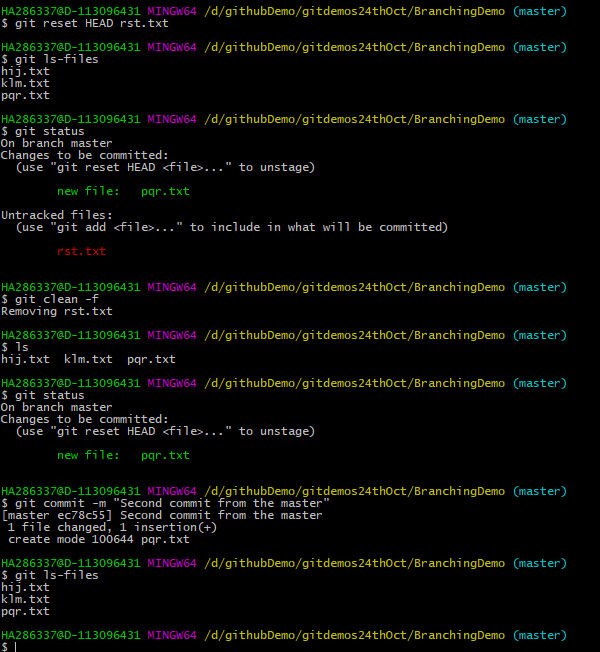


* 1. Working directory vs Local Repository

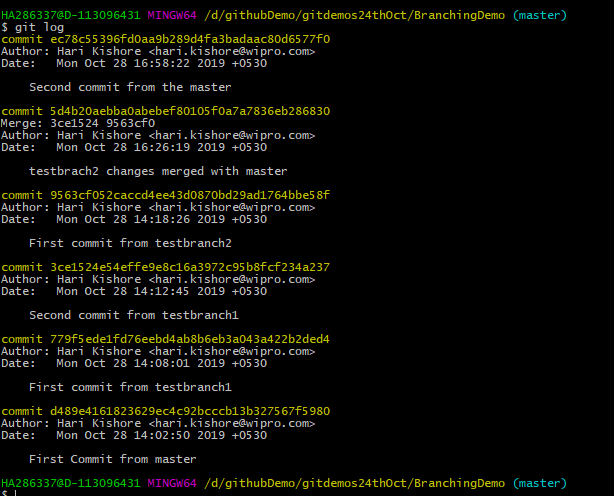


* 1. Staging area vs Local Repository

Before commit, the files will be present in the staging. When we use the git status command to know which files are for the commit.

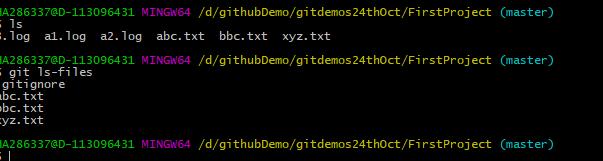


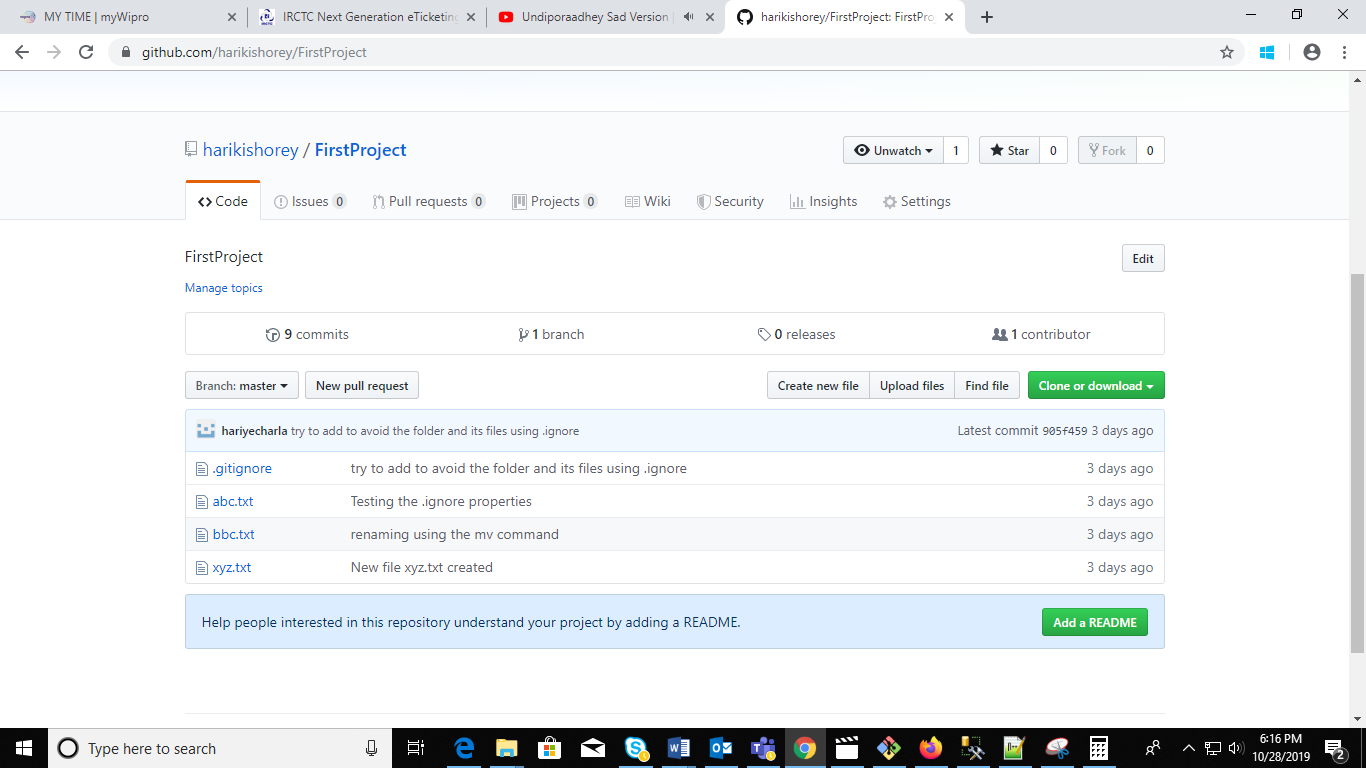
* 1. Between two commits



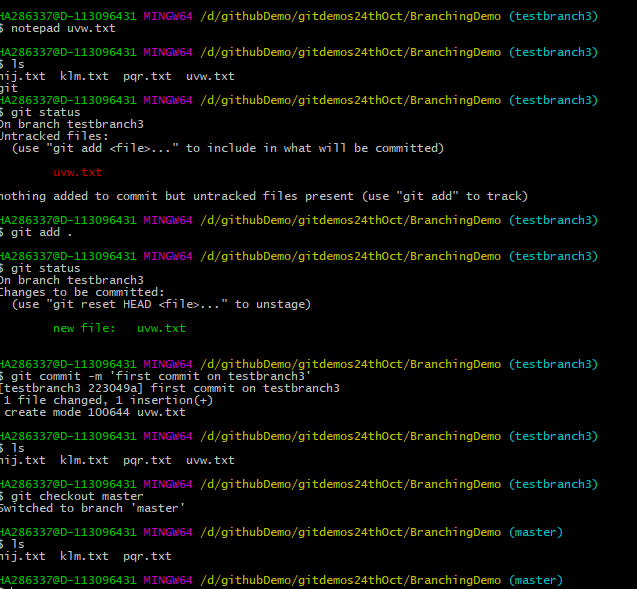
* 1. Between two tags
  2. Local vs Remote Repository

Once we commit the code it will be in the local repository. Once we issuse the push command then only the changes will getting into the remote repository.

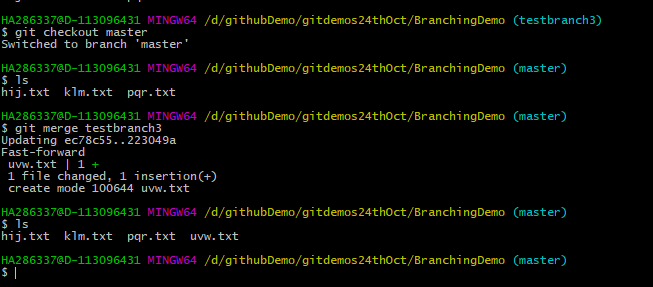




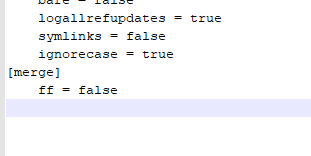
* 1. Master branch vs test branch



1. Merge the changes from test branch to master branch.
   1. FastForward merge

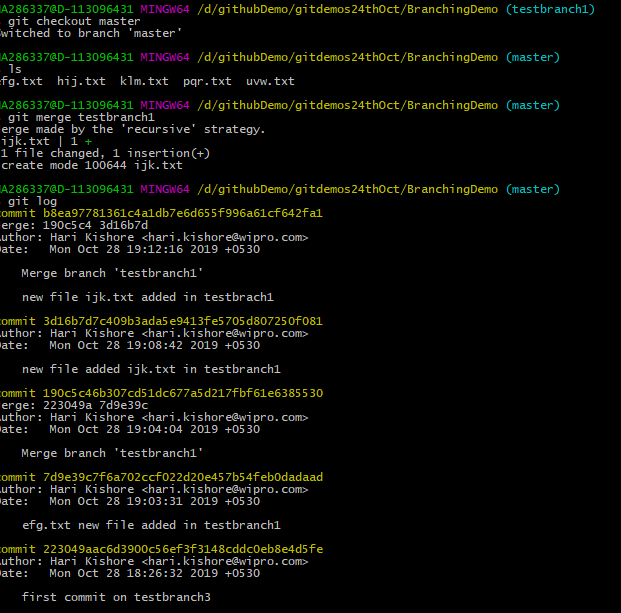


* 1. Disabling FastForward merge

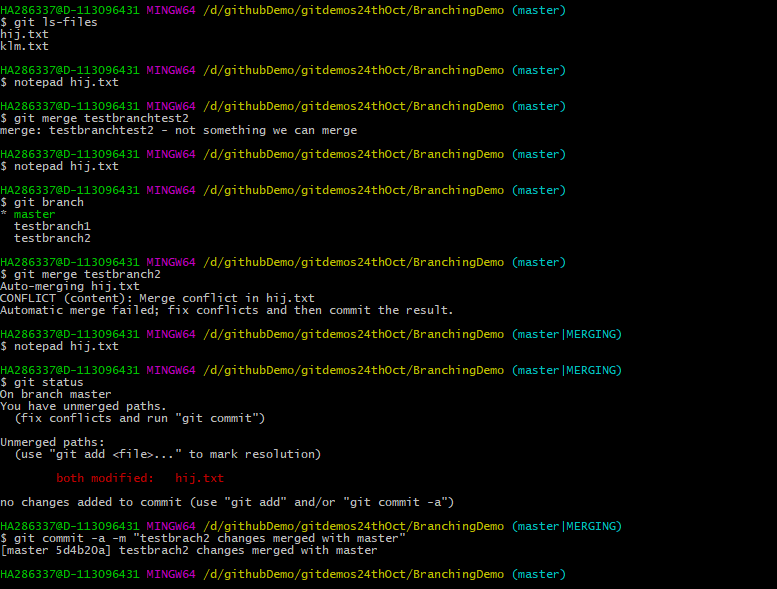


* 1. What is the difference between option a and option b

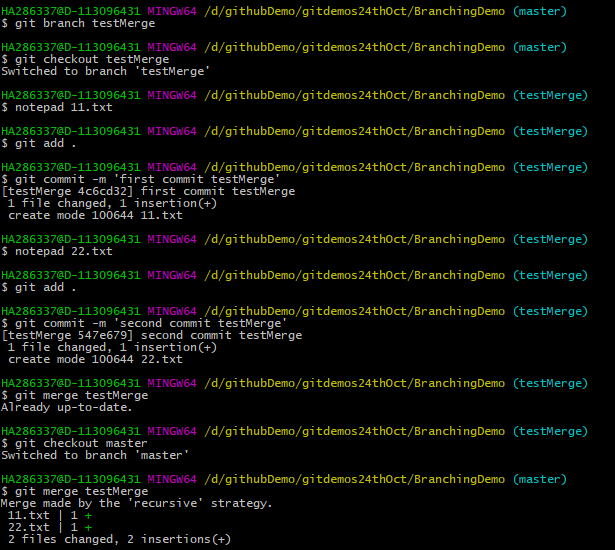
When we use this property, when merge it will add some comments related to the merge.

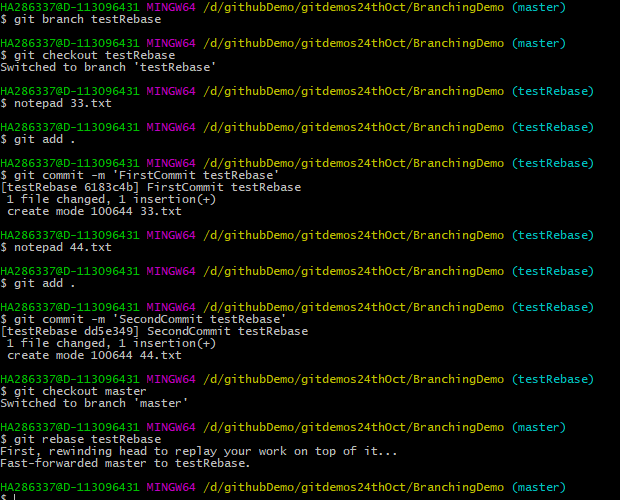


1. Create a merge conflict situation. Resolve the conflict and merge the changes between the branches.



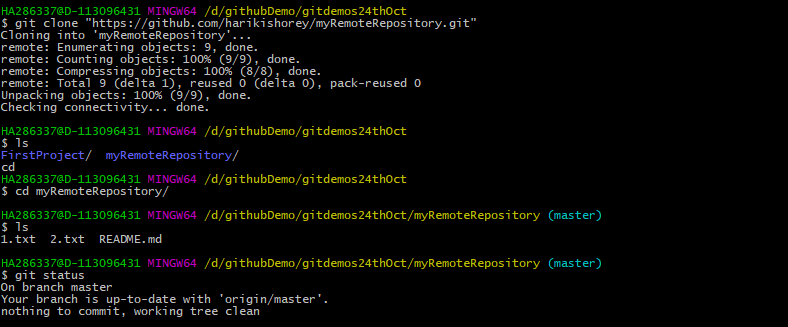
1. What is the difference between merge and rebase, demonstrate with the eg.





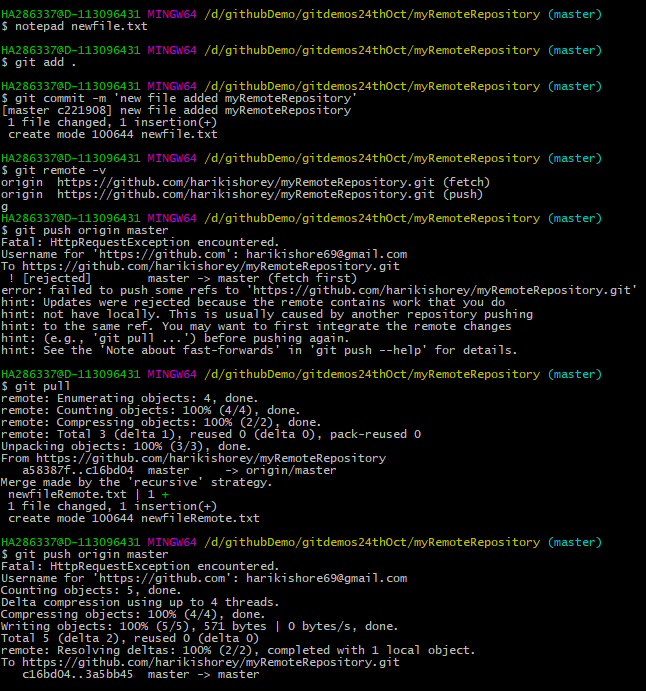
1. With an example, demonstrate fetch, clone and pull. What is the usecase for these operations. Are they same, different? Explain

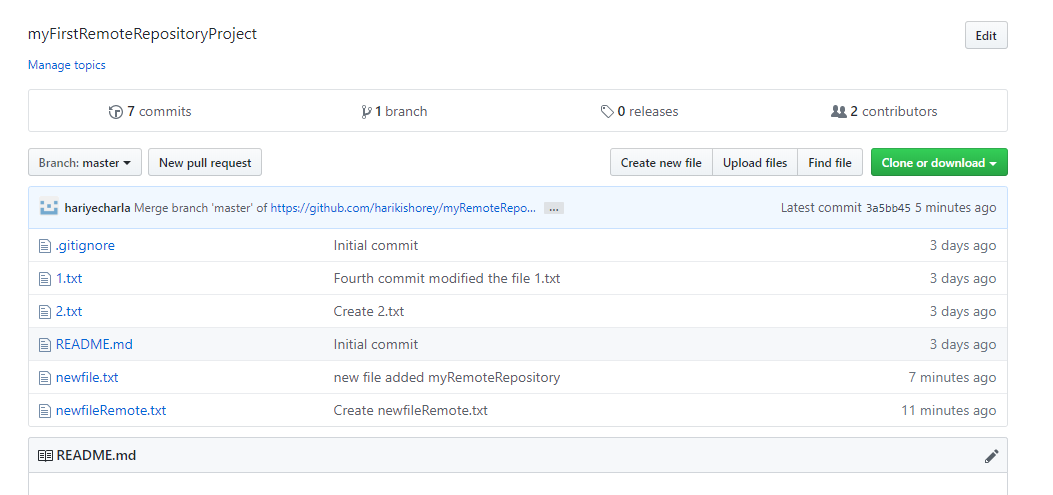
When we have already have the project in the remote repository we can use the command to get the whole remote repository into our local repository git clone command as shown in the below screenshot.



git pull

If we add some files or update an existing files so we need to push these changes to the remote repository mean while any one adds a new file or changes in the existing files it will now allow us to push the changes at that time will use the git pull to get the latest version of the remote repository and then will push the changes to the remote repository.





1. Create a new repository in Github, with a README file. While pushing to the remote repository, if the remote branch is ahead of the local repository (new file is added in remote repository, which is not there in local repository) and pull is failing, how do you solve this problem?

There is not issues. I have created a remote project by selecting public and readme file and added a file. Then created a local git repository and created one file there. So before push the changes to the remote server I executed the pull command get the latest version from the remote then I pushed the changes successfully.

