## **Session : Introduction to Version Control**

## **EXERCISE**

**By: Syed Shayan Nasir (EmployeeID: 4153)**

1. Git Setup https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html

Ans. **sudo apt-get install git**

**git –version**

**git config –global user.email “**[**sshayann7@gmail.com**](mailto:sshayann7@gmail.com)**”**

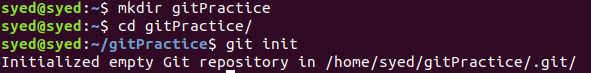
**git config –global user.name “shayannasir”**

1. Initialize a Git Repository

Ans. **mkdir gitPractice**

**cd gitPractice/**

**git init**



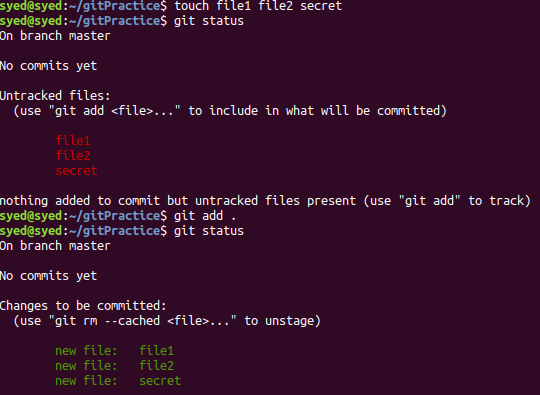
1. Add files to the repository

Ans. **touch file1 file2 secret**

**git status**

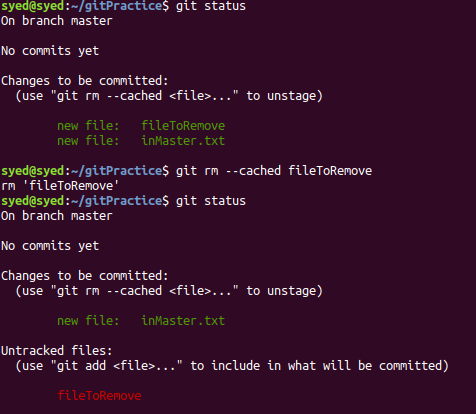
**git add .**

**git status**



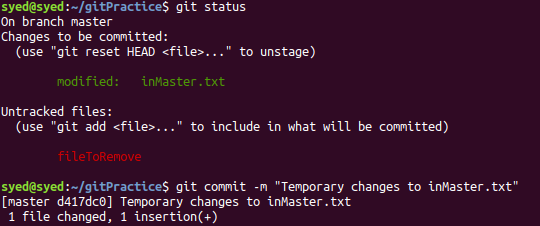
1. Unstage 1 file

Ans. **git rm –cached fileToRemove**



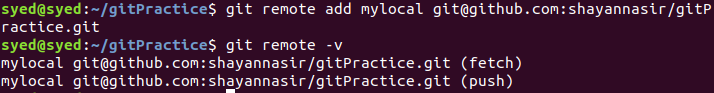
1. Commit the file

Ans. **git commit -m “Temporary changes to inMaster.txt”**



1. Add a remote

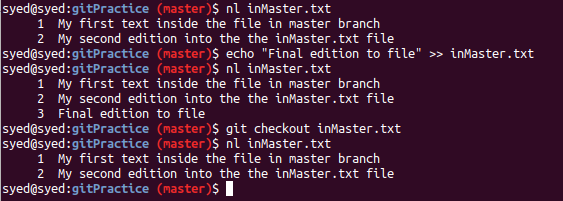
Ans. **git remote add mylocal git@github.com:shayannasir/gitPractice.git**



1. Undo changes to a particular file

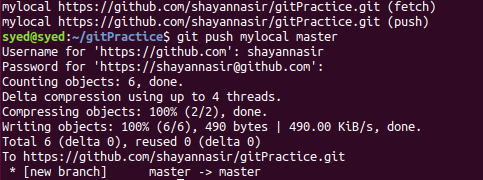
Ans. **echo "Final edition to file" >> inMaster.txt**

**git checkout inMaster.txt**



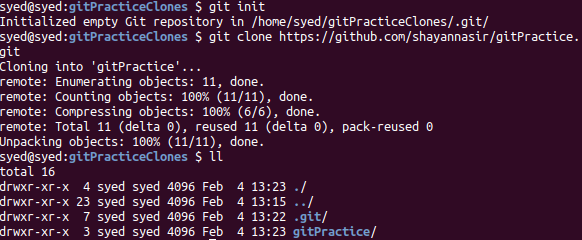
1. Push changes to Github

Ans. **git push mylocal master**



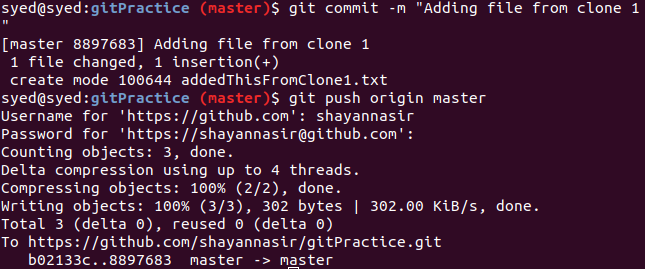
1. Clone the repository

Ans. **git clone** [**https://github.com/shayannasir/gitPractice.git**](https://github.com/shayannasir/gitPractice.git)

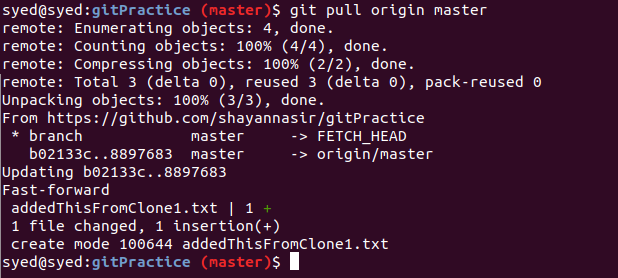


1. Add changes to one of the copies and pull the changes in the other.

Ans. **Adding changes and pushing from clone1**



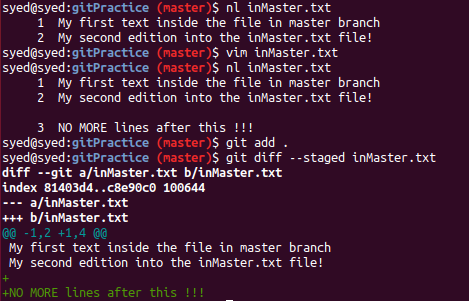
**Pulling the clone1 made changes from github to clone2**



1. Check differences between a file and its staged version

Ans. **git add .**

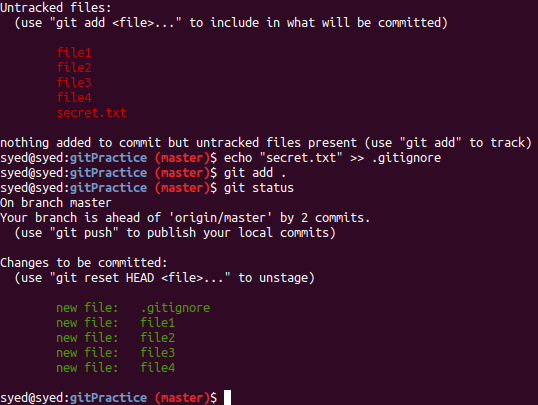
**git diff --staged inMaster.txt**



1. Ignore a few files to be checked in

Ans. **echo "secret.txt" >> .gitignore**

**git add .**

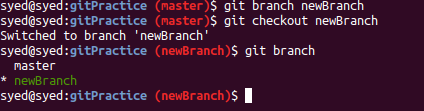


1. Create a new branch.

Ans. **git branch newBranch**

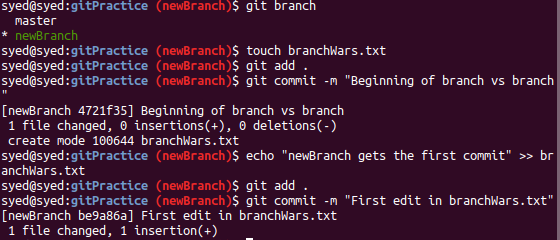
**git checkout newBranch**

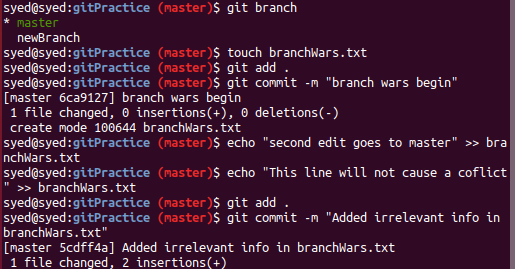
**git branch**



1. Diverge them with commits

Ans. Moving newBranch ahead with two commits

Diverging master branch with commits



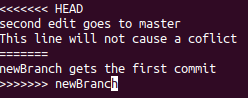
1. Edit the same file at the same line on both branches and commit

Ans. Covered in the code snippets of the previous questions. **First line of branchWars.txt in both the branches have been edited and committed**

1. Try merging and resolve merge conflicts



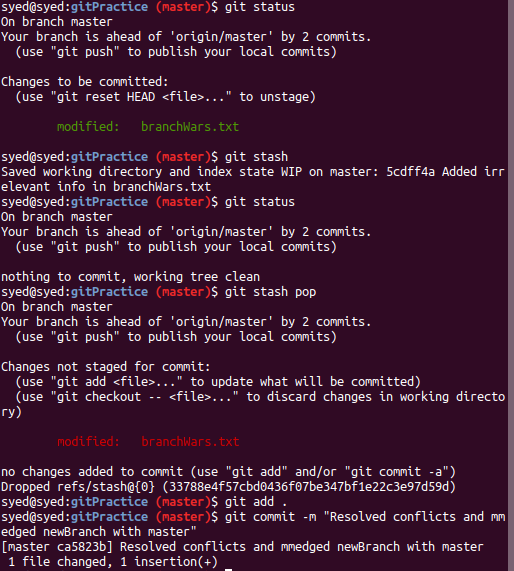








1. Stash the changes and pop them



1. Add the following code to your .bashrc file : color\_prompt="yes"  
   parse\_git\_branch() {  
   git branch 2> /dev/null | sed -e '/^[^\*]/d' -e 's/\* \(.\*\)/(\1)/'  
   }  
   if [ "$color\_prompt" = yes ]; then  
   PS1='\u@\h\[\033[00m\]:\[\033[01;34m\]\W\[\033[01;31m\] $(parse\_git\_branch)\[\033[00m\]\$ '  
   else  
   PS1='\u@\h:\W $(parse\_git\_branch)\$ '  
   fi  
   unset color\_prompt force\_color\_prompt

Ans. **Step 1: Copy code**

**Step 2: sudo vim ~/.bashrc**

**Step 3: Write and quit**

**Step 4: Observe the change in color and the addition of branch name in the Terminal Prompt**