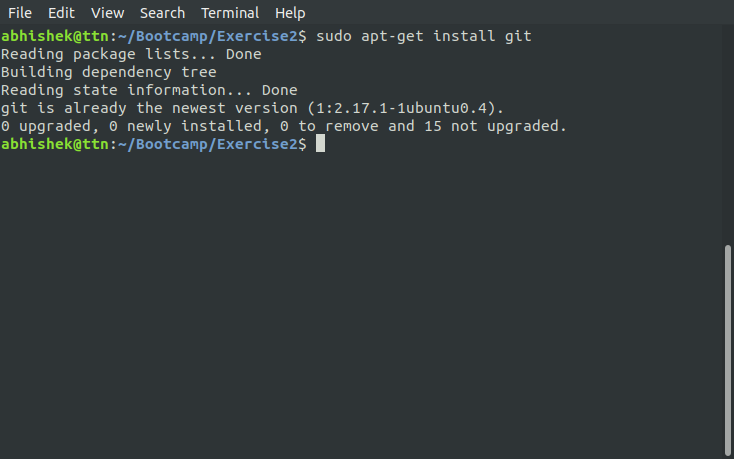
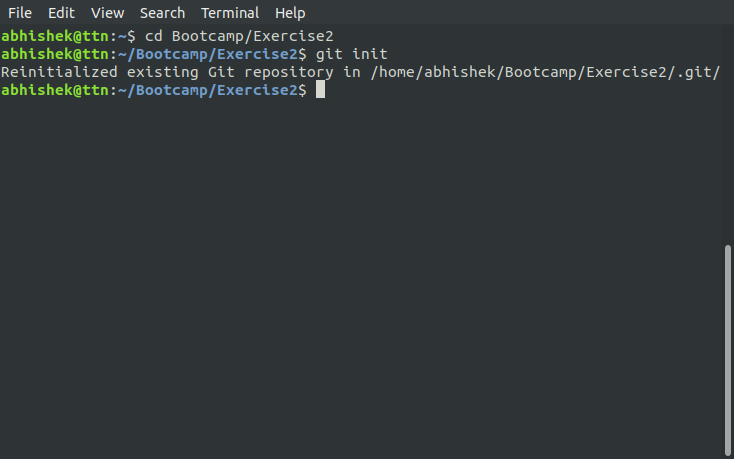
# Version control & Git Exercise 2 Abhishek Maurya

1. **Git Setup** [**https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html**](https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html)
   * sudo apt-get install git

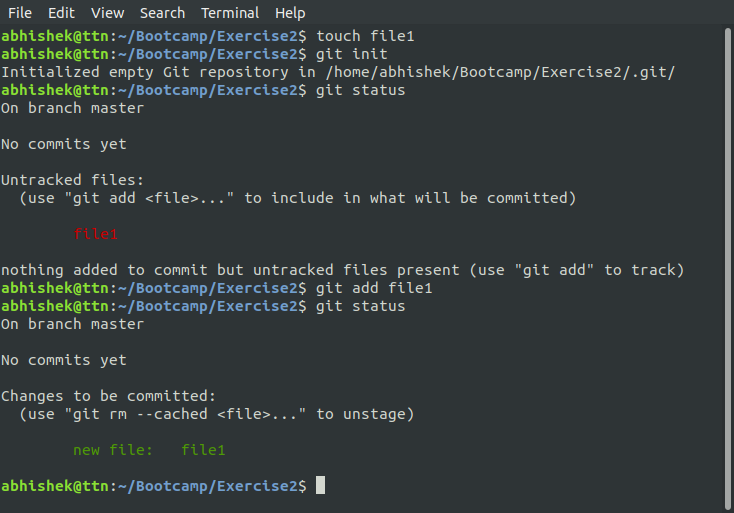


1. **Initialize a Git Repository**
   * git init

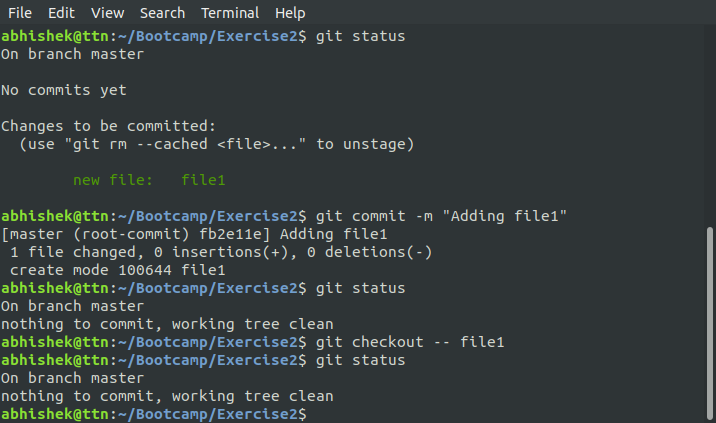


1. **Add files to the repository**
   * touch file1

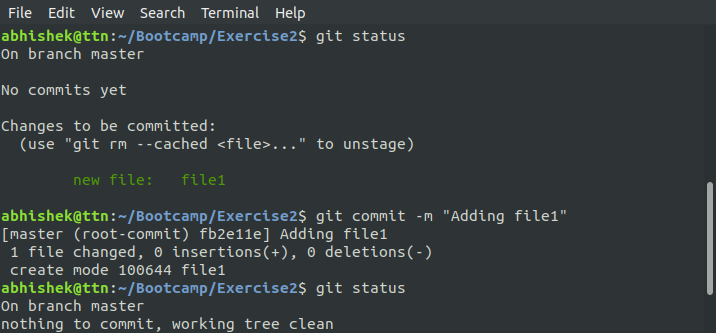
git add file1



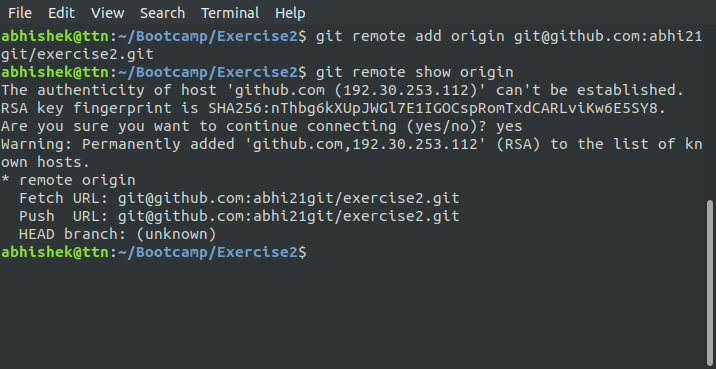
1. **Unstage 1 file**
   * git reset HEAD file1



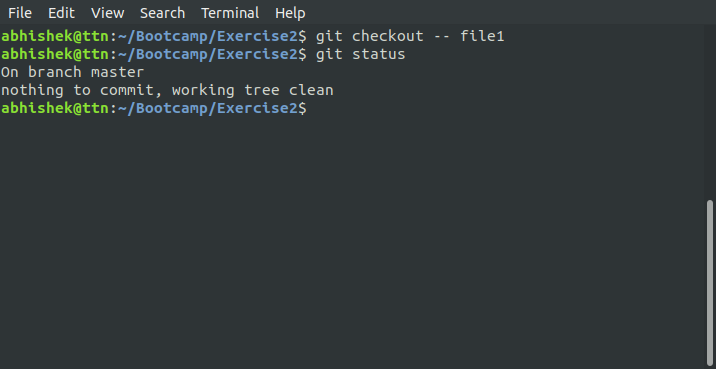
1. **Commit the file**
   * git commit -m "Adding file1"



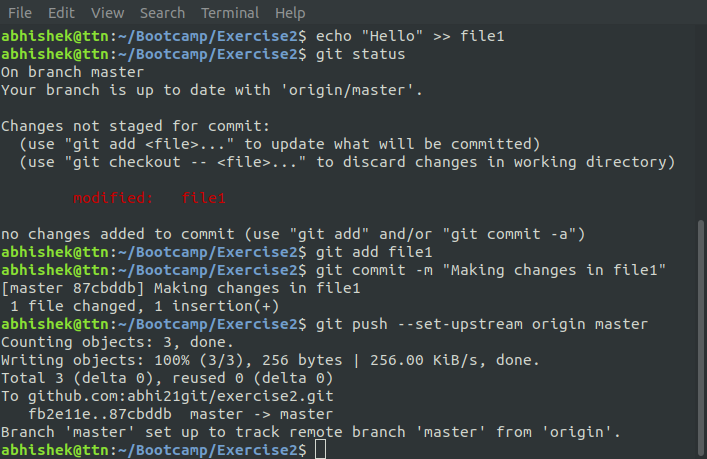
1. **Add a remote**
   * git remote add origin git@github.com:abhi21git/exercise2.git



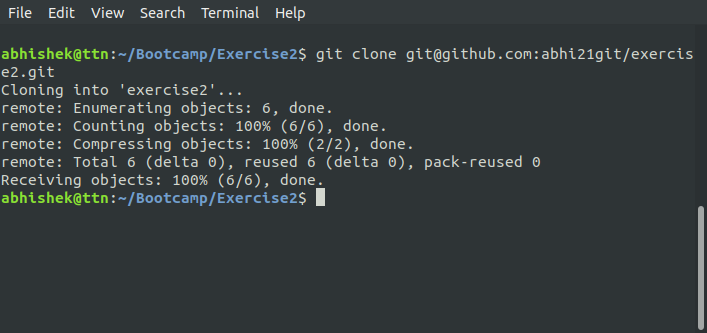
1. **Undo changes to a particular file**
   * git checkout -- file1



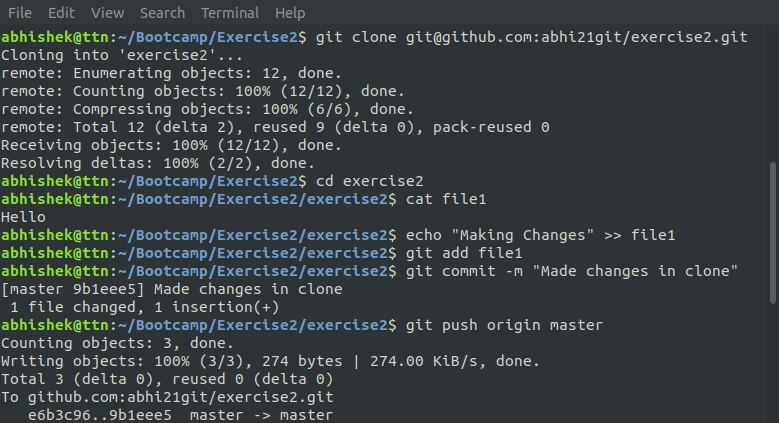
1. **Push changes to Github**
   * git push --set-upstream origin master

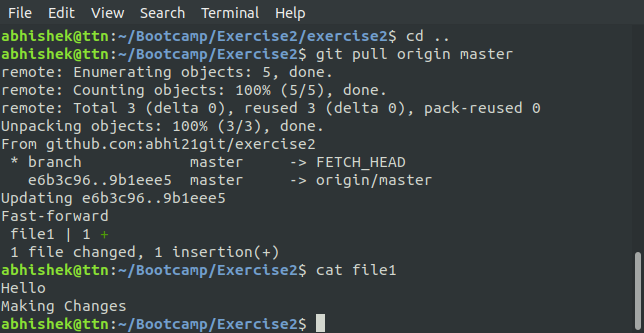


1. **Clone the repository**
   * git clone git@github.com:abhi21git/exercise2.git

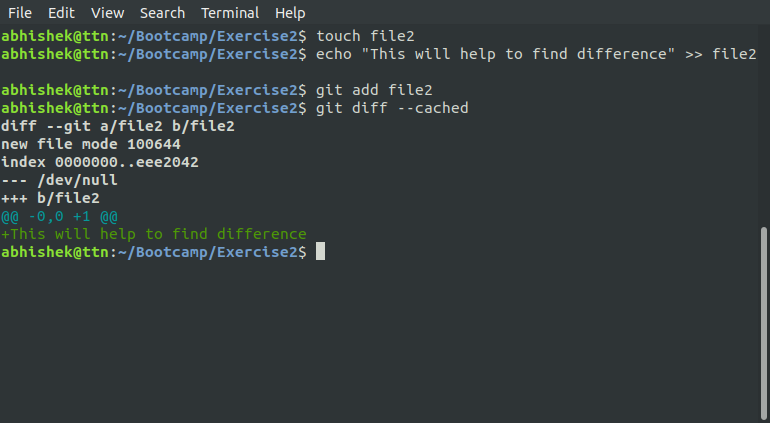


1. **Add changes to one of the copies and pull the changes in the other.**
   * echo "Making changes" >> exercise2/file1 //at clone directory
   * git pull origin master //at original directory

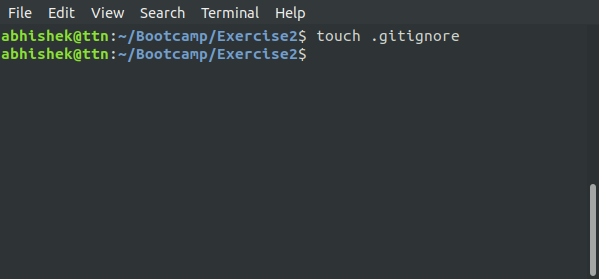




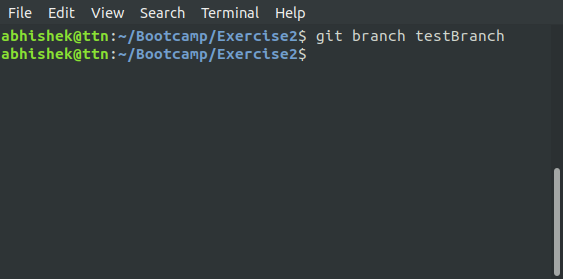
1. **Check differences between a file and its staged version**
   * git diff --cached



1. **Ignore a few files to be checked in**
   * touch .gitignore



1. **Create a new branch.**
   * git branch testBranch



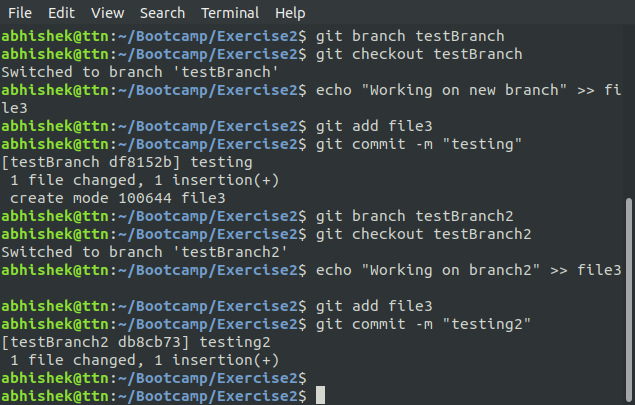
1. **Diverge them with commits**
   * git branch testBranch

git checkout testBranch

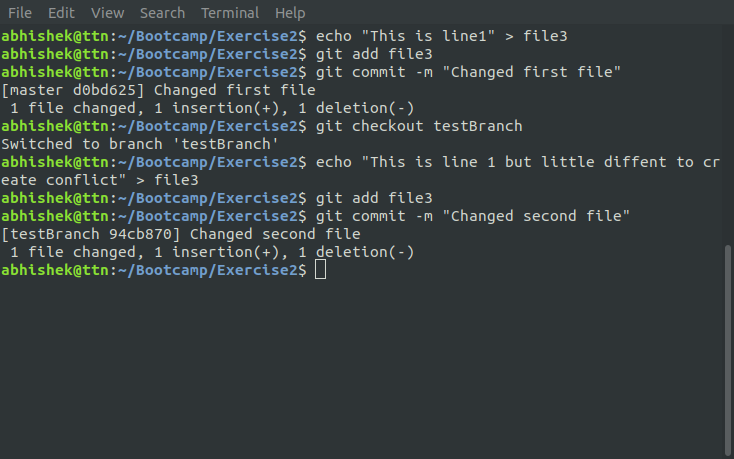
echo "Working on test branch" >> file3

git add file3

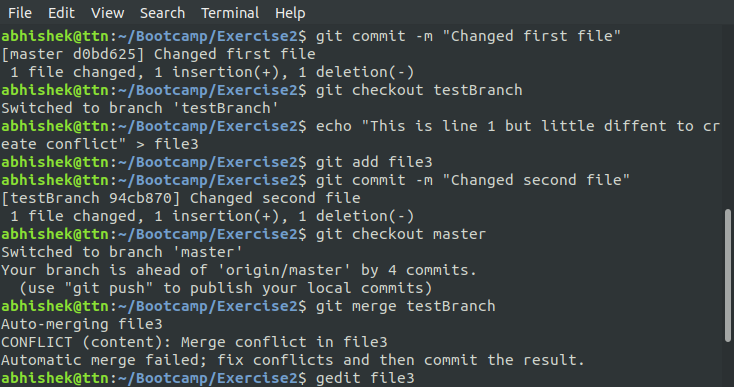
git commit -m "testing"



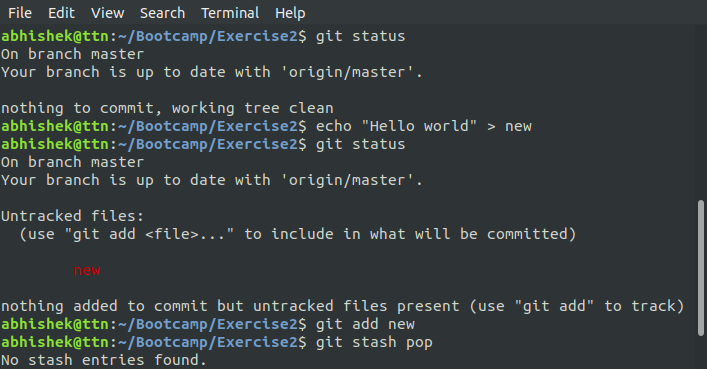
1. **Edit the same file at the same line on both branches and commit**



1. **Try merging and resolve merge conflicts**
   * git merge testBranch



1. **Stash the changes and pop them**
   * git stash pop



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**

