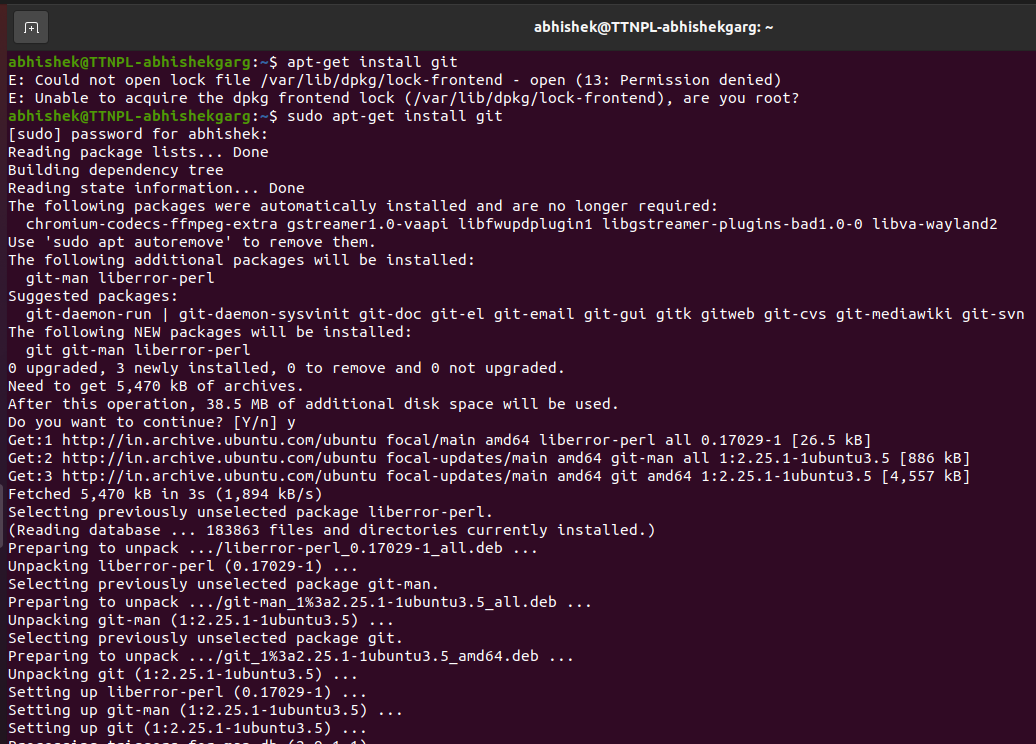
Exercise: Introduction to Version Control

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

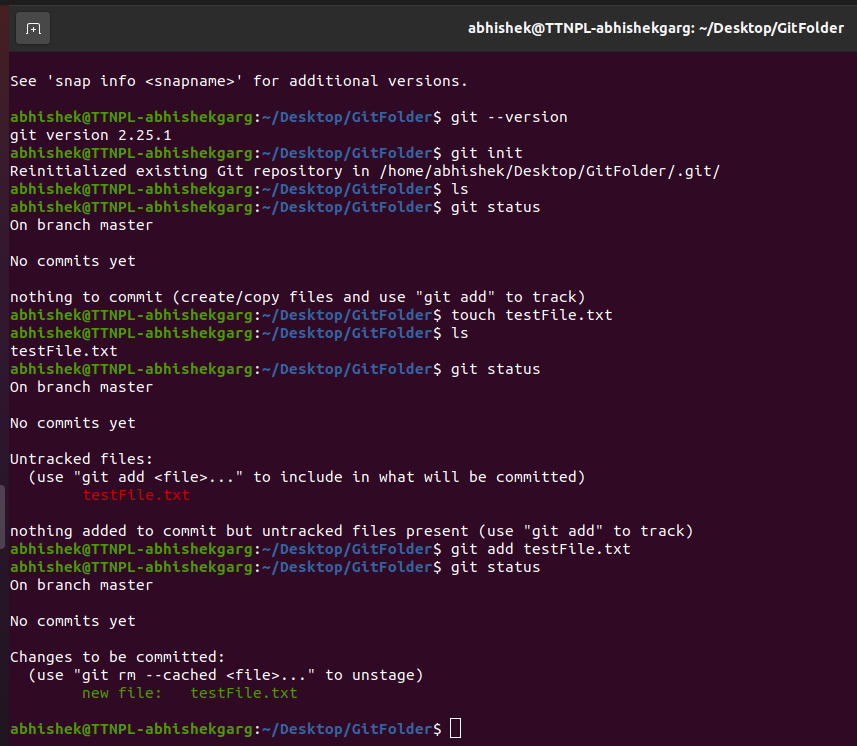
Answer: sudo apt-get install git



1. Initialize a Git Repository

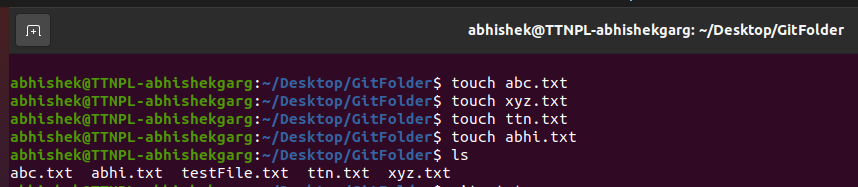
Answer: Navigate to the desired directory and enter command:-

Git init



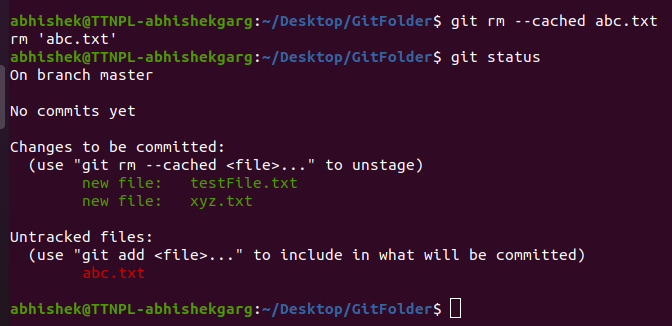
1. Add files to the repository

Answer: touch abc.txt xyz.txt ttn.txt abhi.txt testFile.txt



1. Unstage 1 file

Answer: git rm - -cached abc.txt



1. Commit the file

Answer: git config - -global user.email “[abhishek.garg@tothenew.com](mailto:abhishek.garg@tothenew.com)”

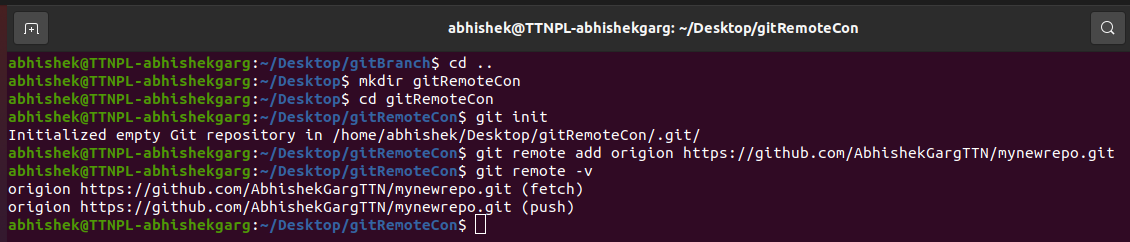
git config - -global user.name “Abhishek”

git commit -m “Initial Commit”



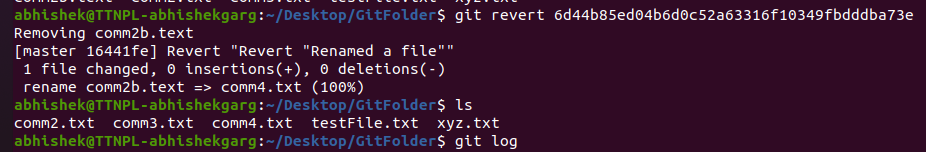
1. Add a remote

Answer: git remote add origin https://github.com/AbhishekGargTTN/mynewrepo.git



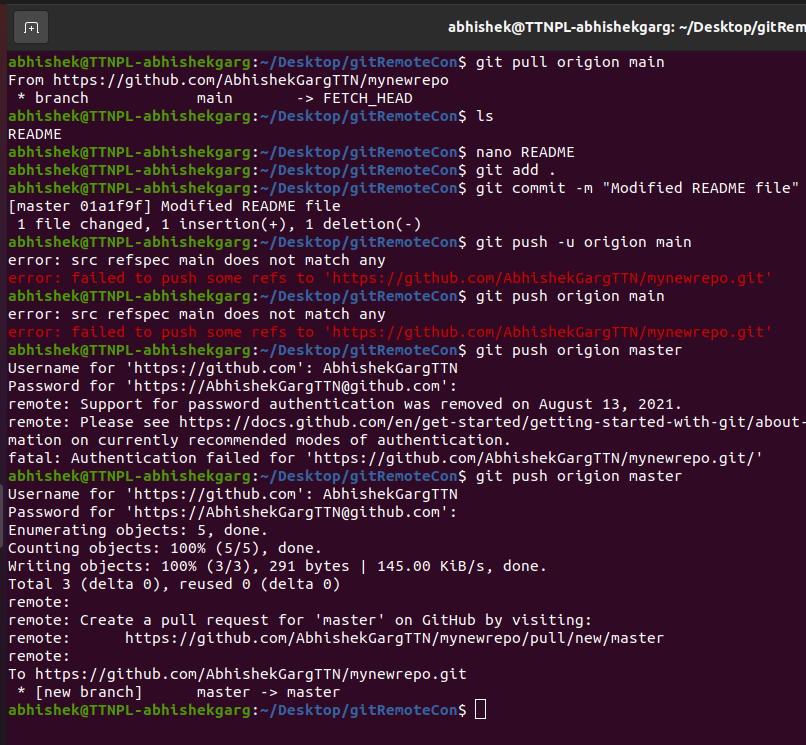
1. Undo changes to a particular file

Answer: git revert 6d44b85



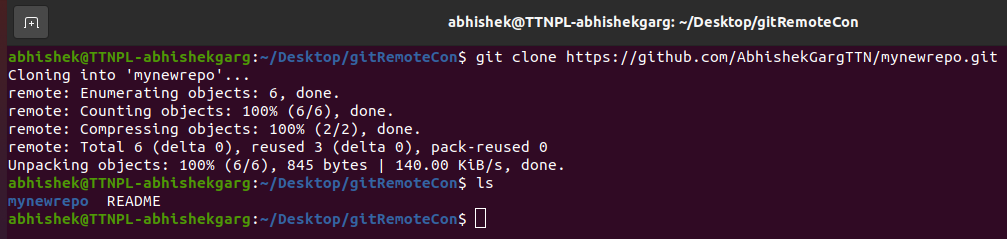
1. Push changes to Github

Answer: git push origin master



1. Clone the repository

Answer: git clone https://github.com/AbhishekGargTTN/mynewrepo.git

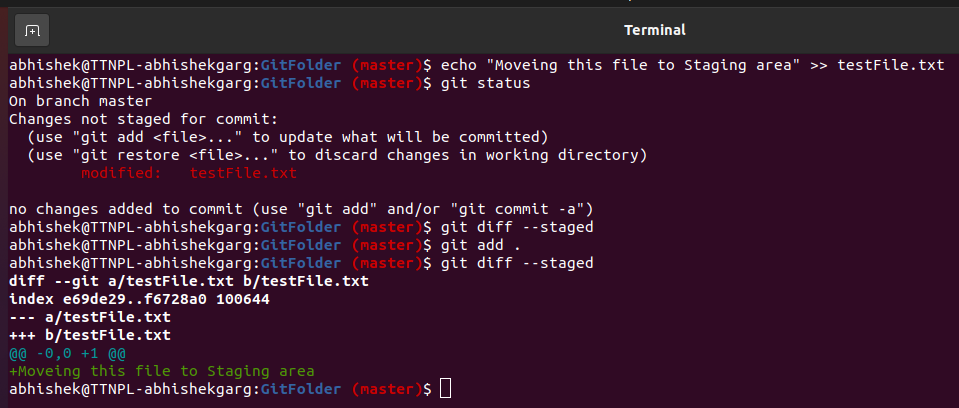


1. Add changes to one of the copies and pull the changes in the other.

Answer:

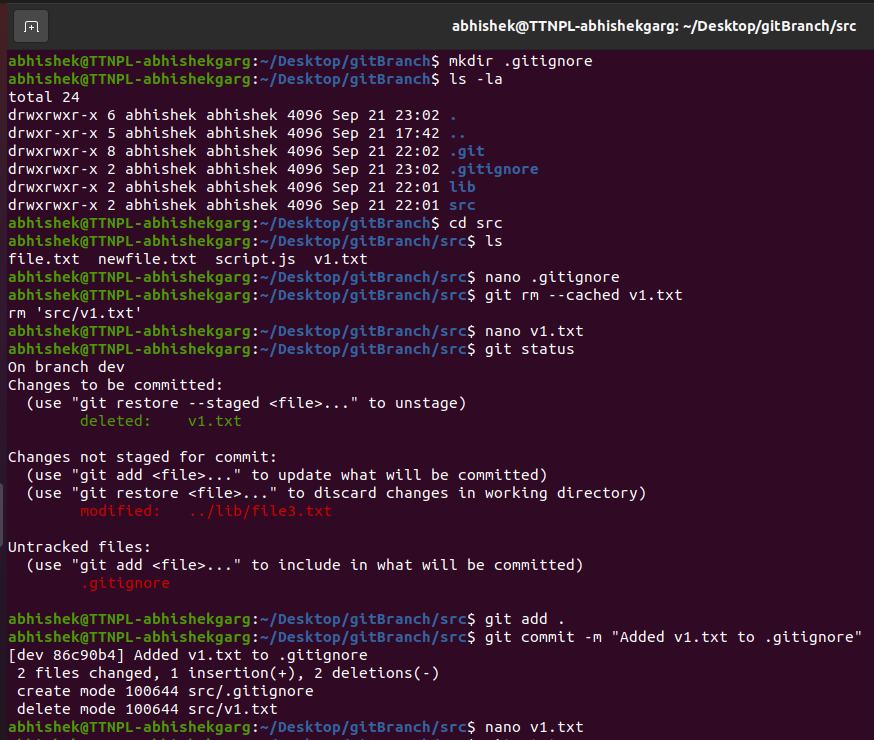
1. Check differences between a file and its staged version

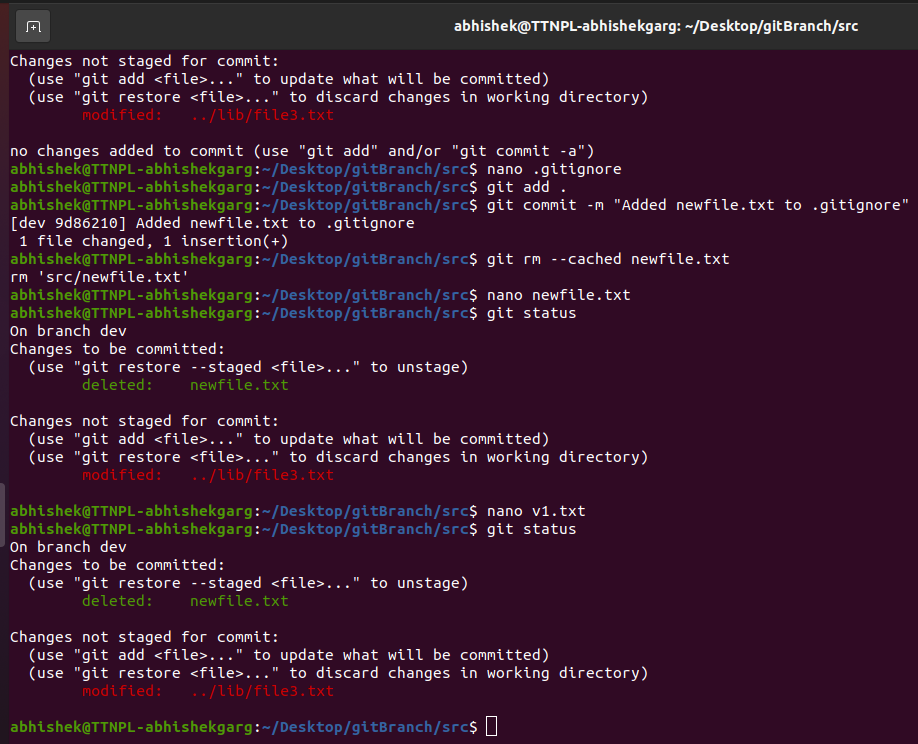
Answer: git diff --staged



1. Ignore a few files to be checked in

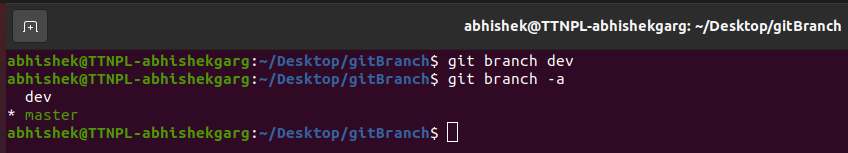
Answer: Create directory .gitignore and add file names to .gitignore





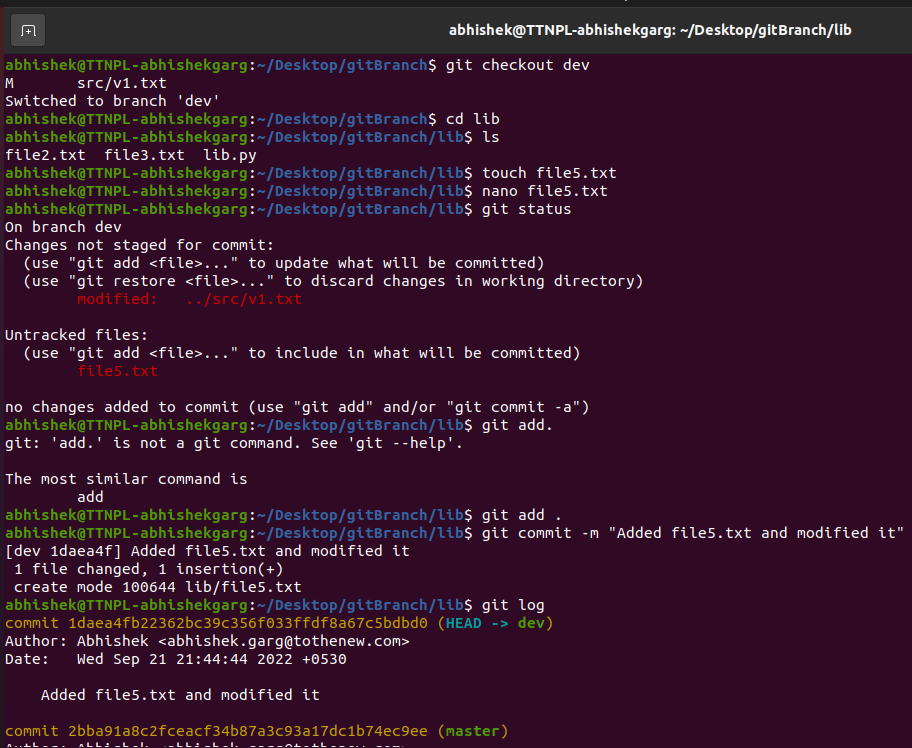
1. Create a new branch.

Answer: git branch dev



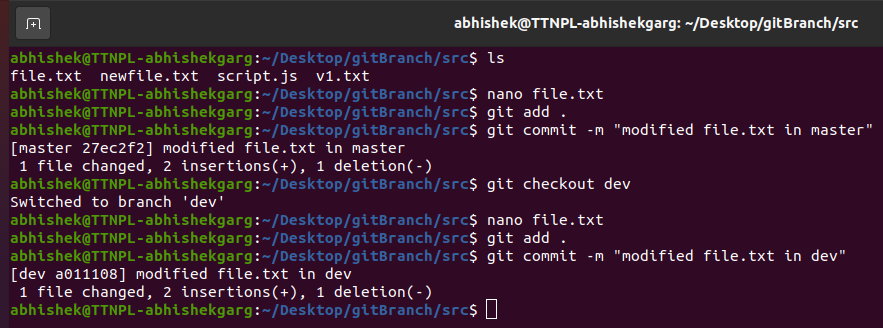
1. Diverge them with commits

Answer: Do some changes to dev branch and commit it



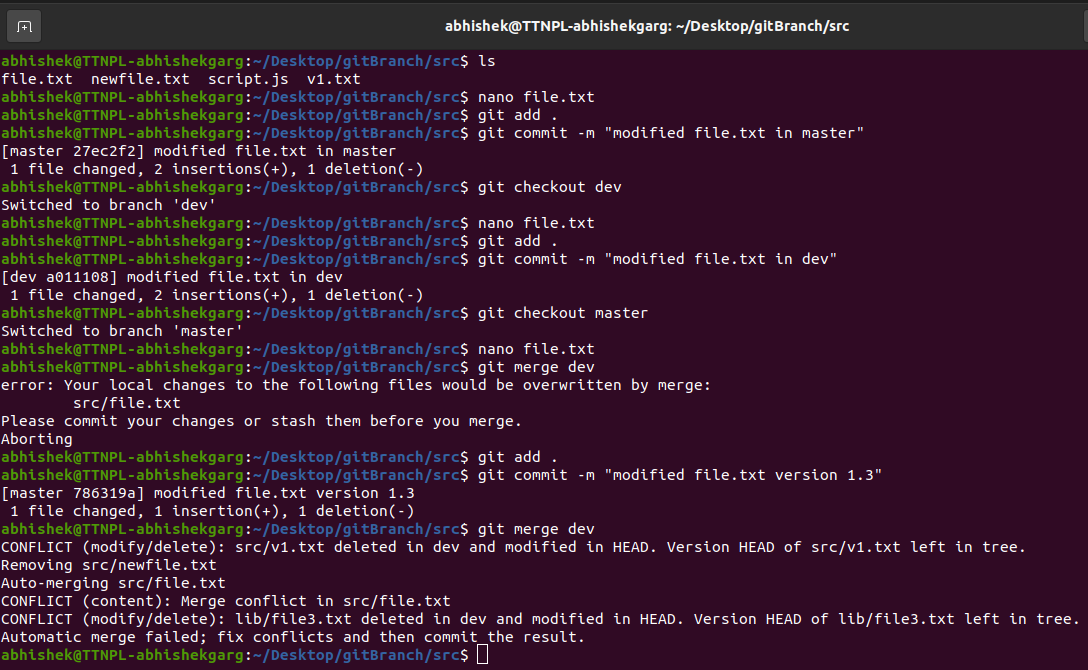
1. Edit the same file at the same line on both branches and commit

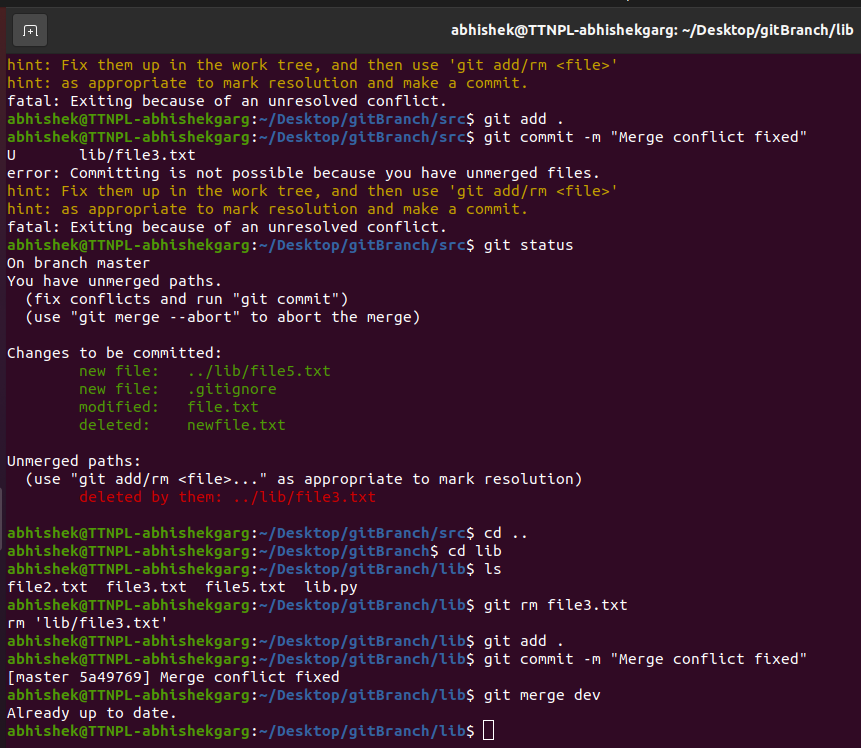
Answer: Edit file.txt in master, commit it and then edit file.txt in dev, commit it



1. Try merging and resolve merge conflicts

Answer:



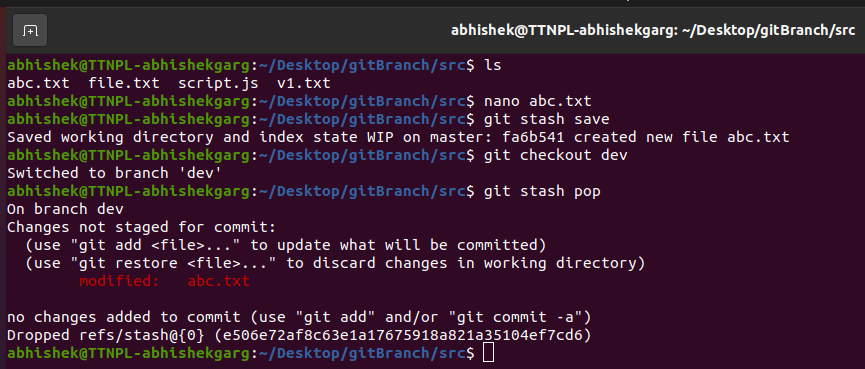


1. Stash the changes and pop them

Answer: git stash save

git checkout dev

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

Answer: nano ~/.bashrc

Copy the code above and paste in .bashrc file and save

