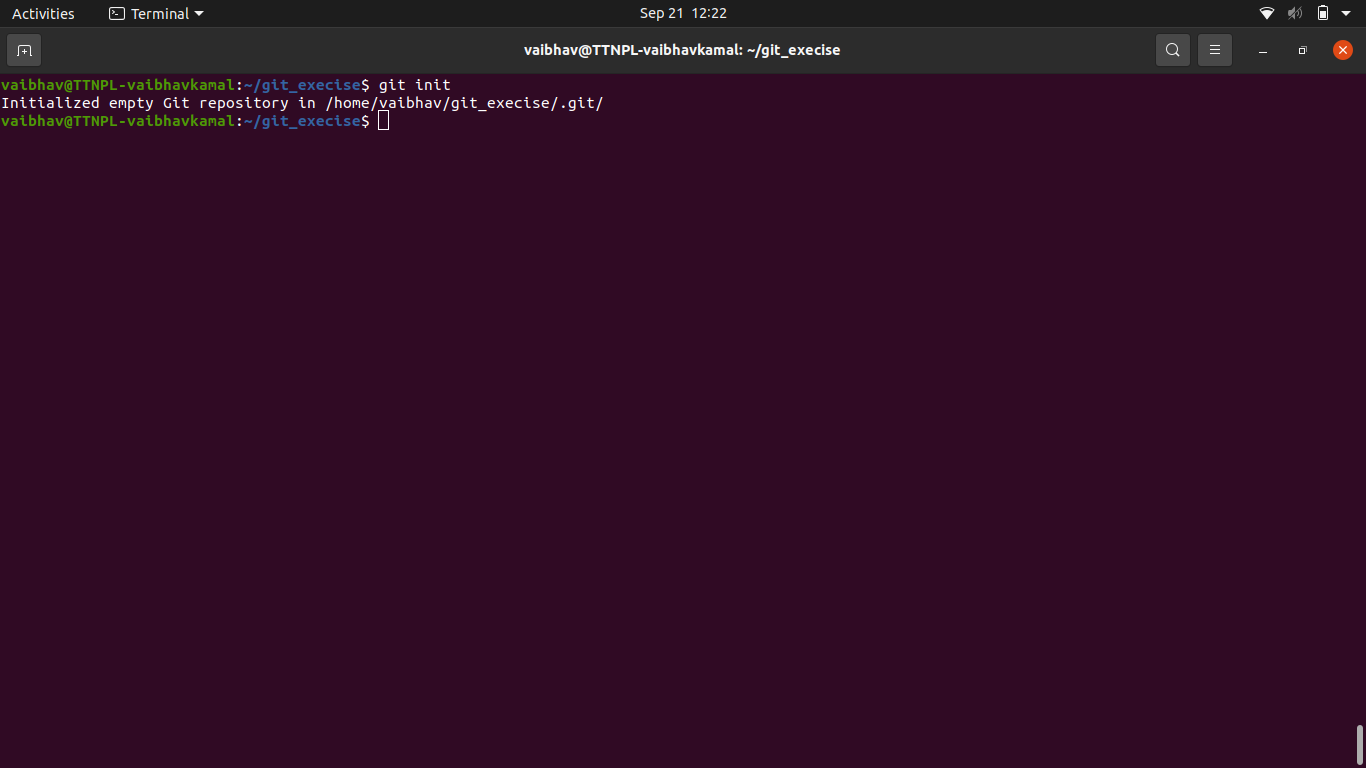
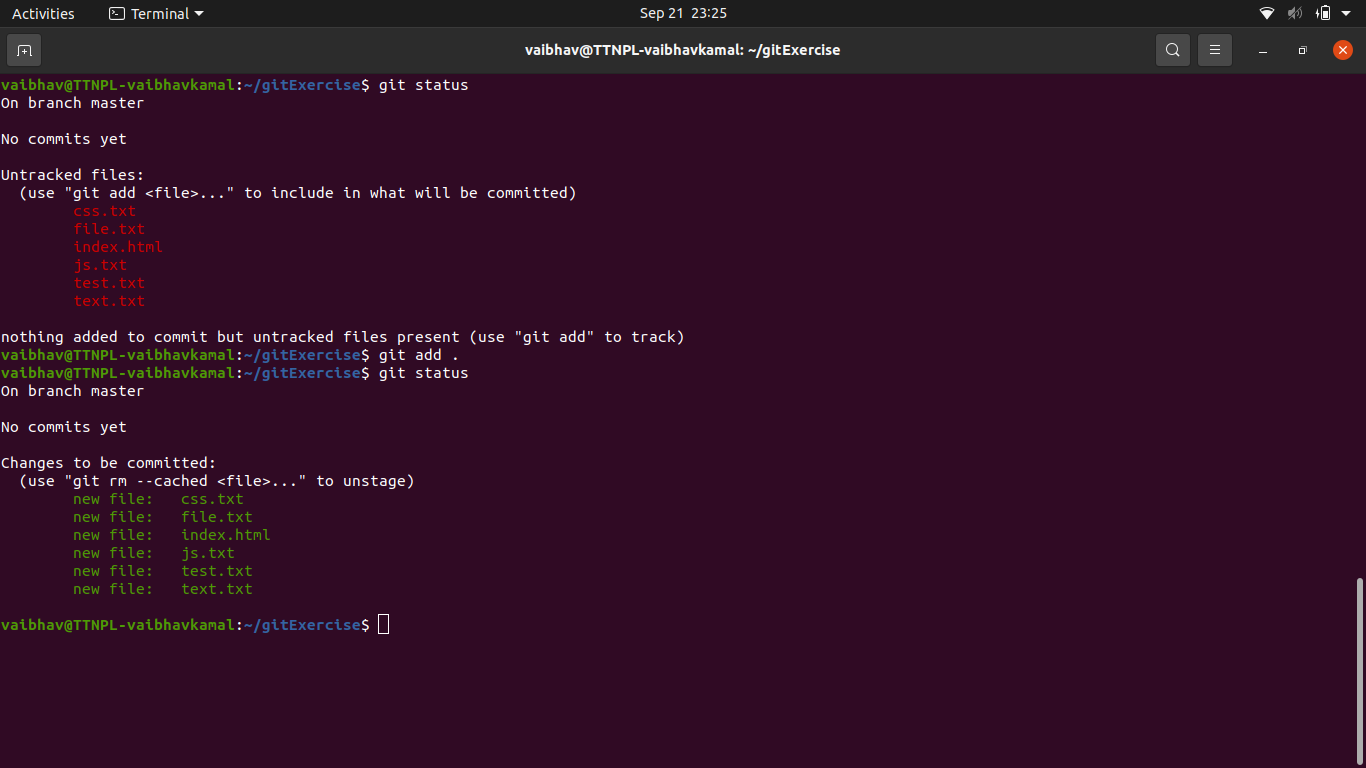
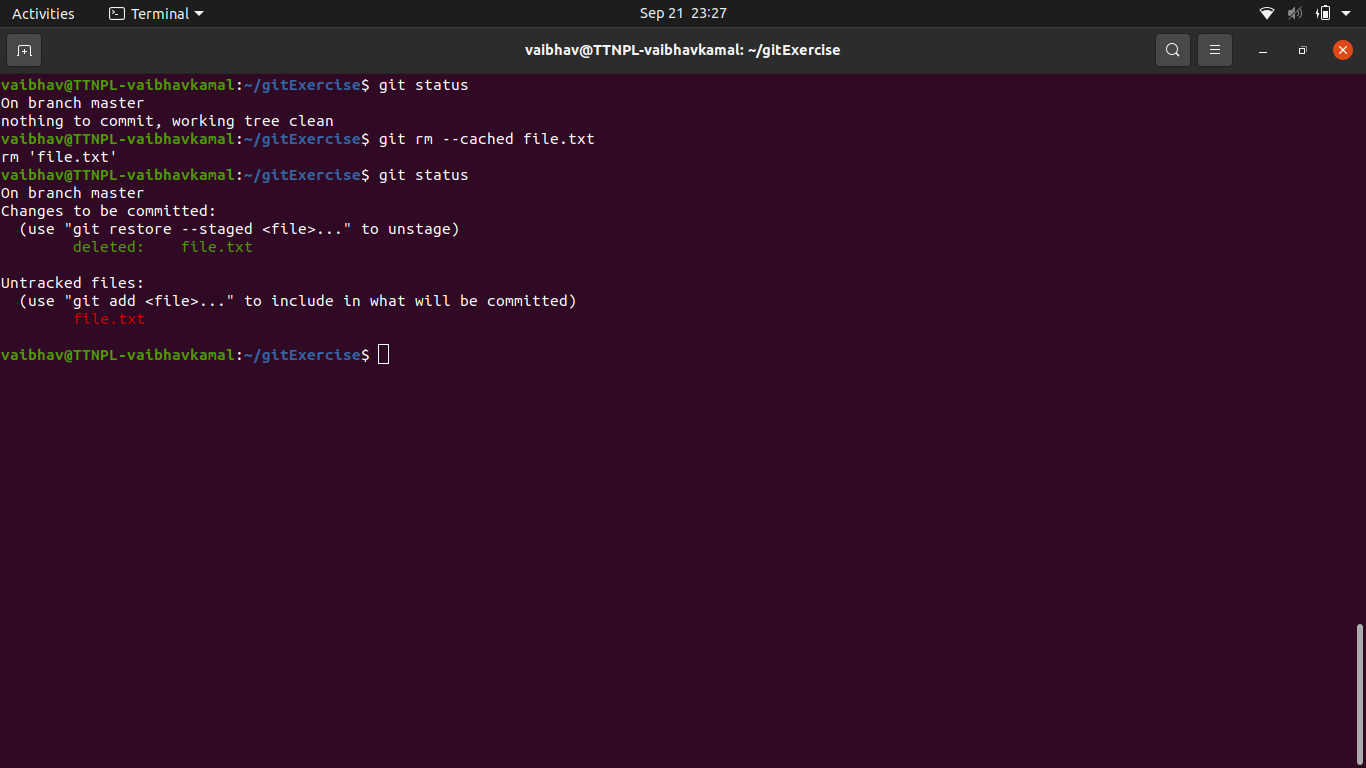
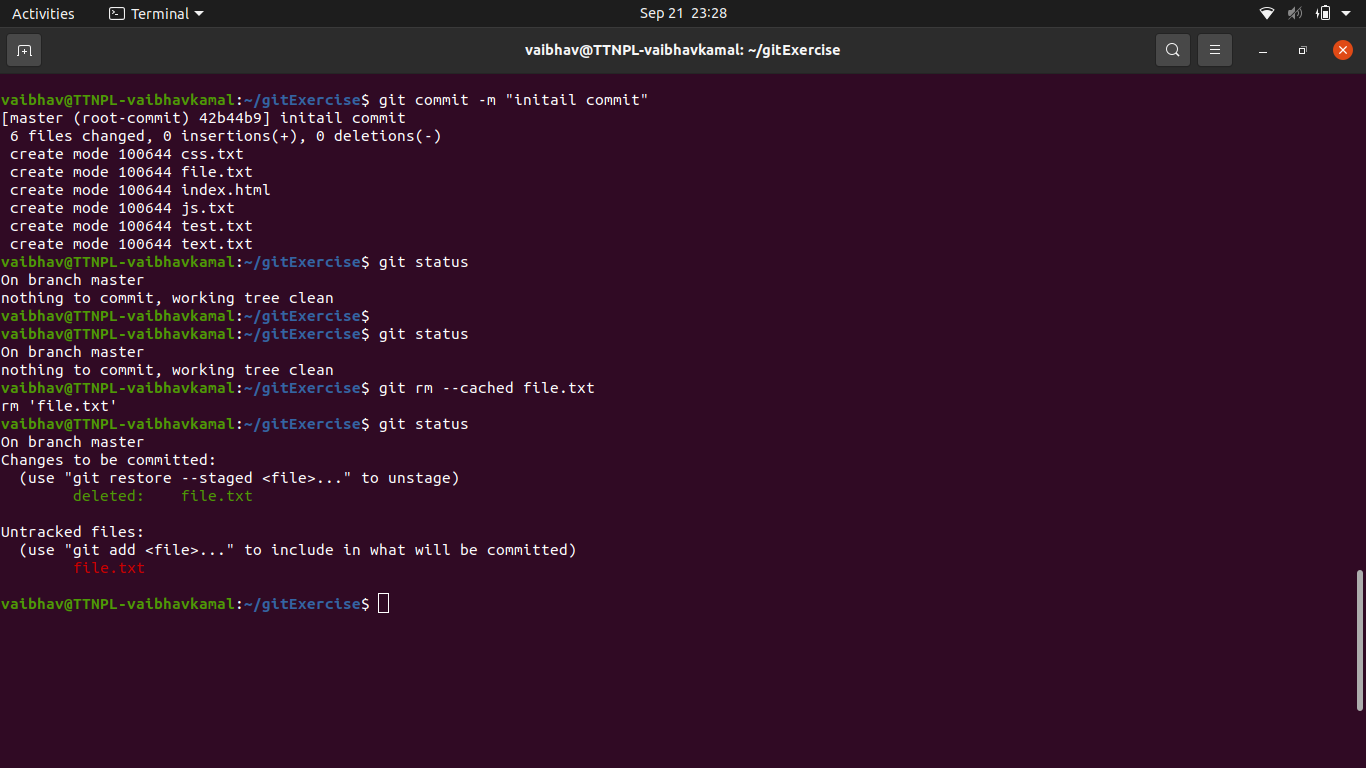
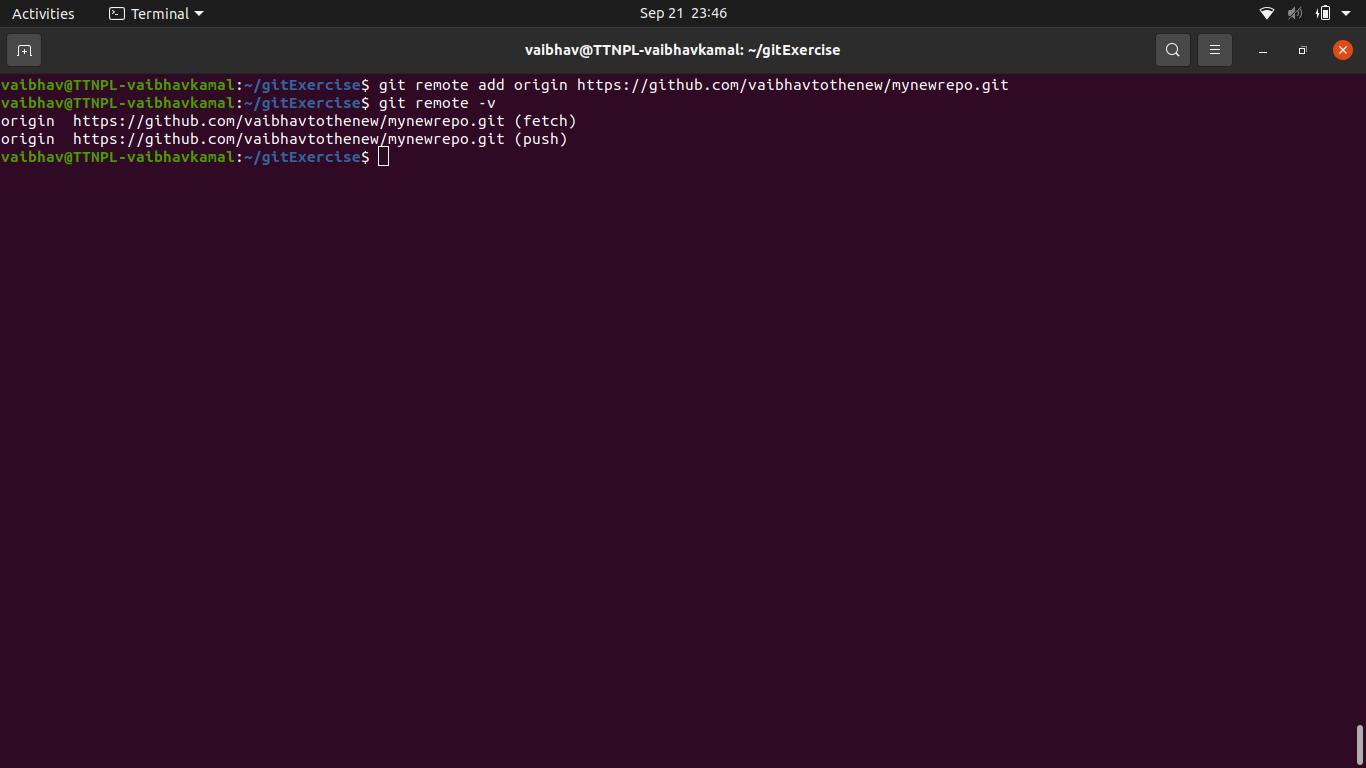
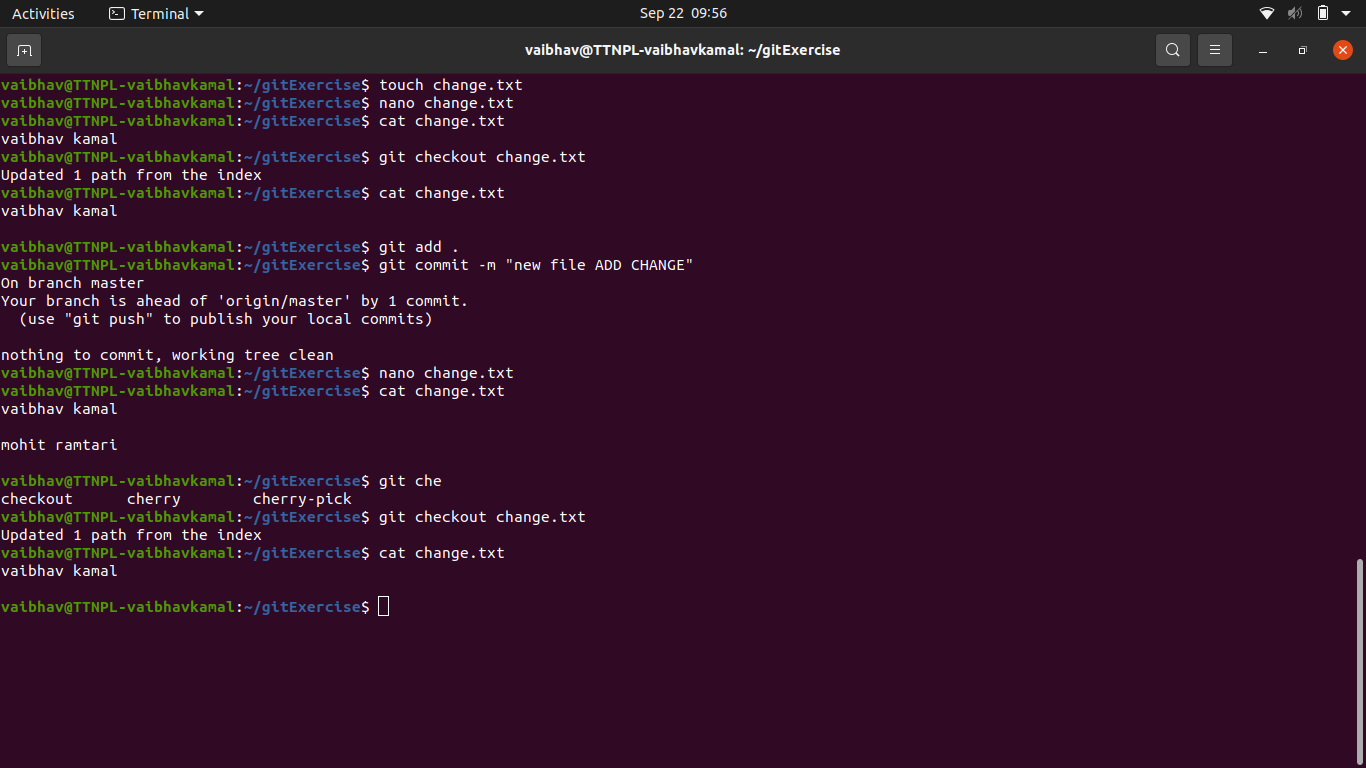
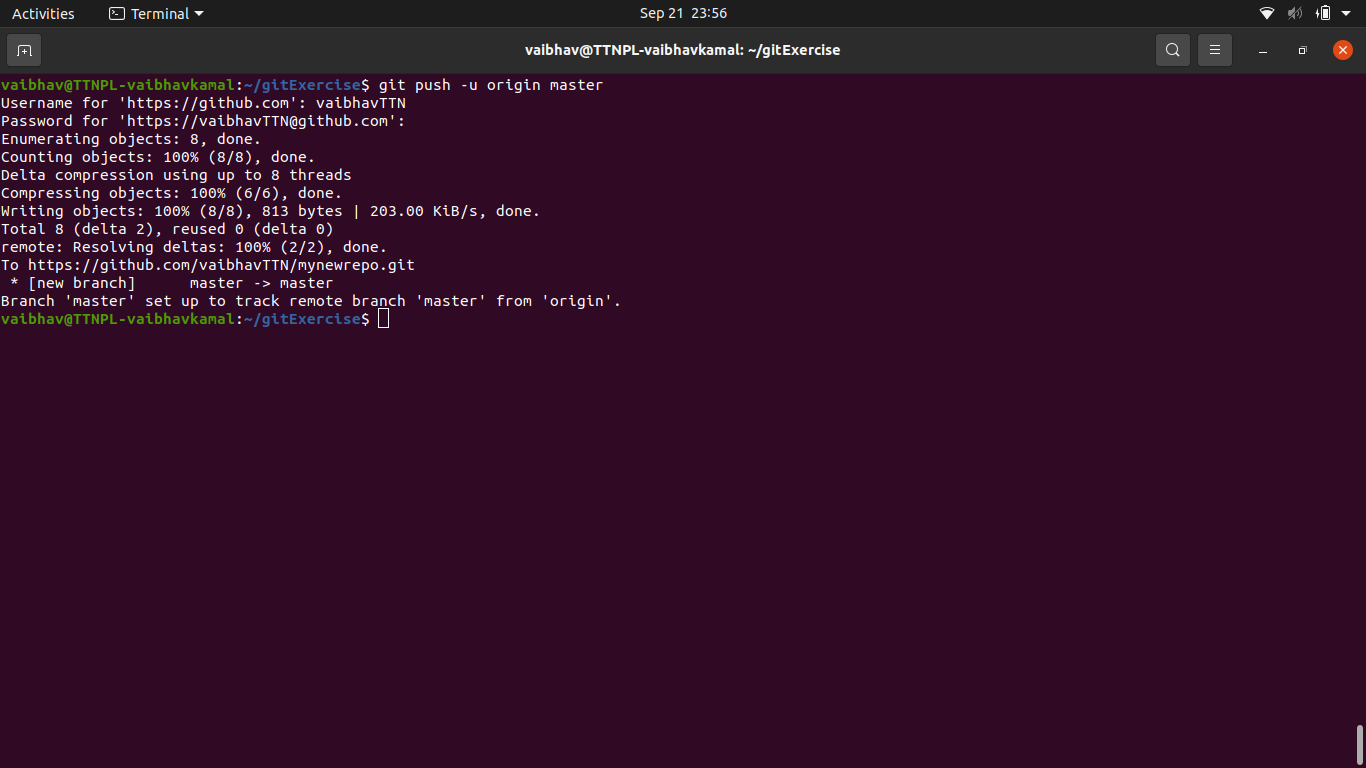
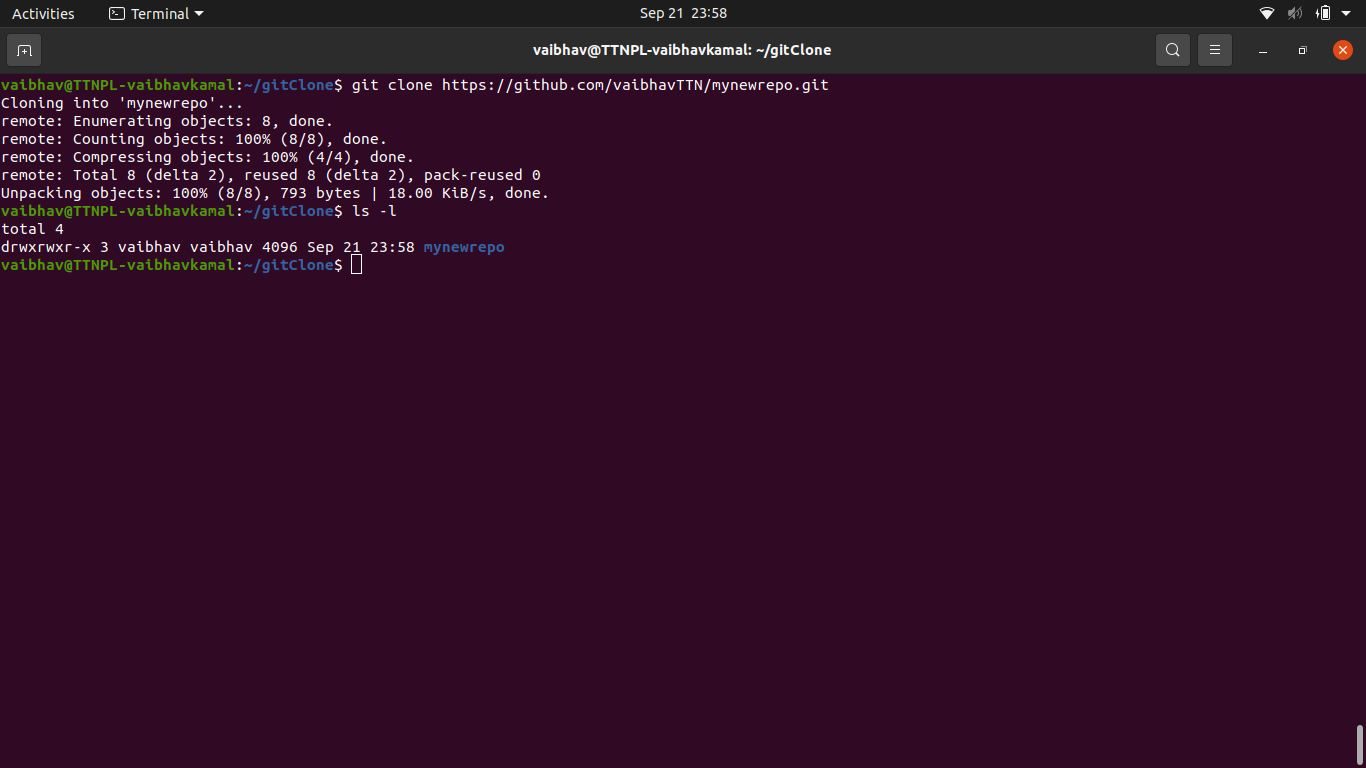
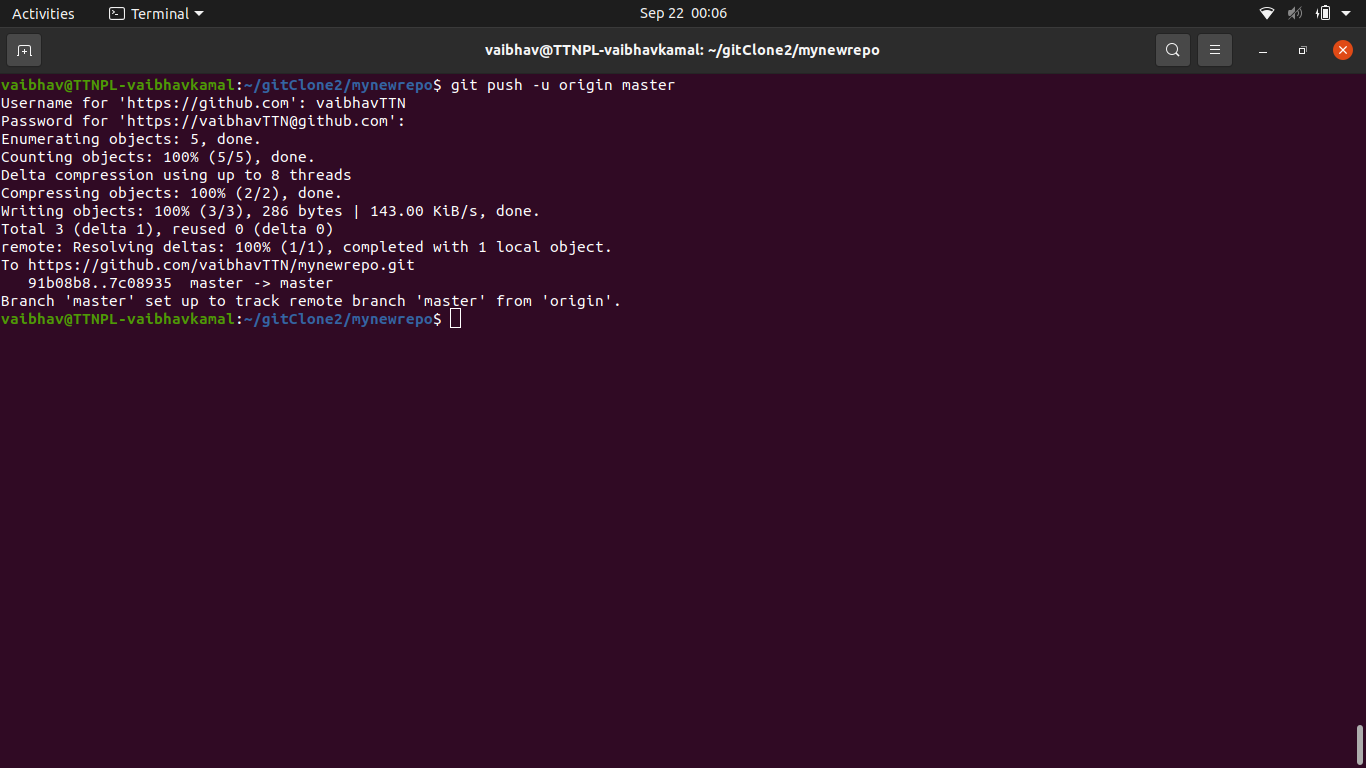
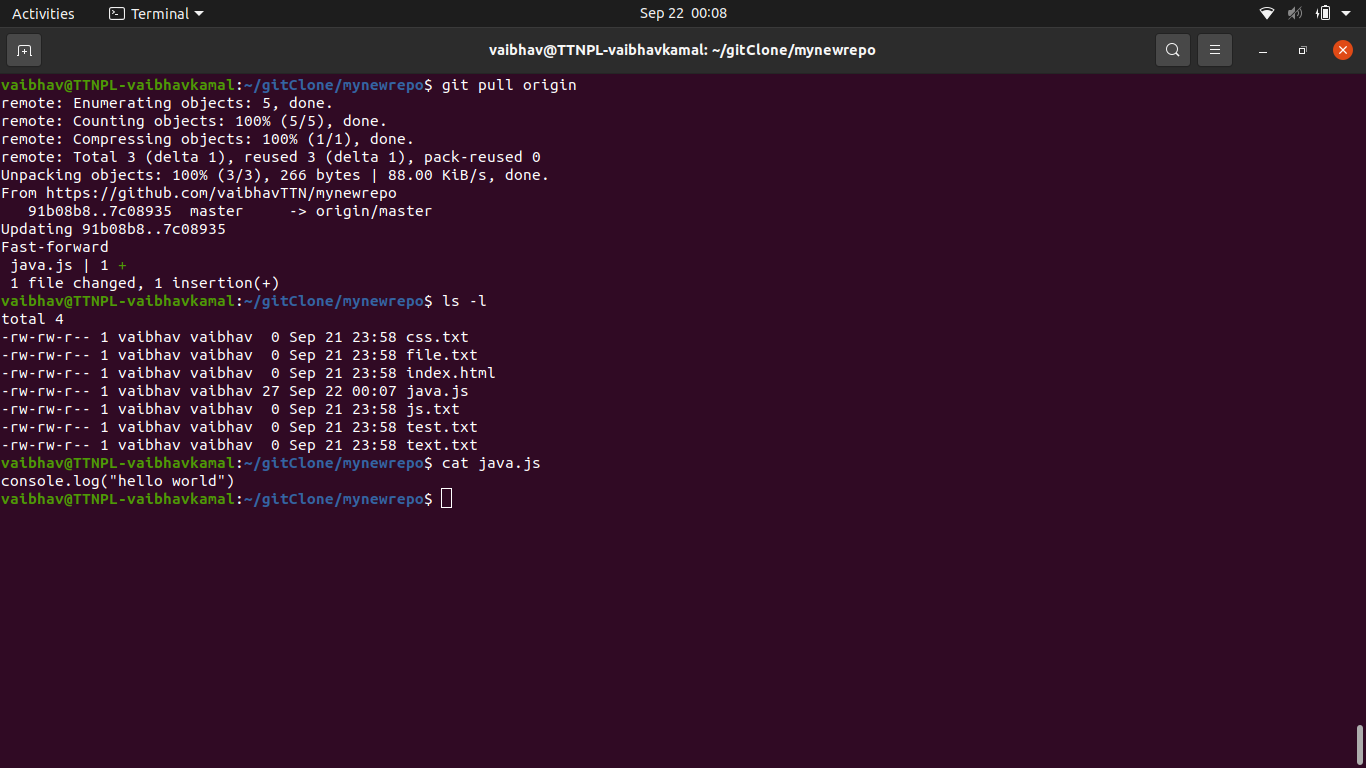
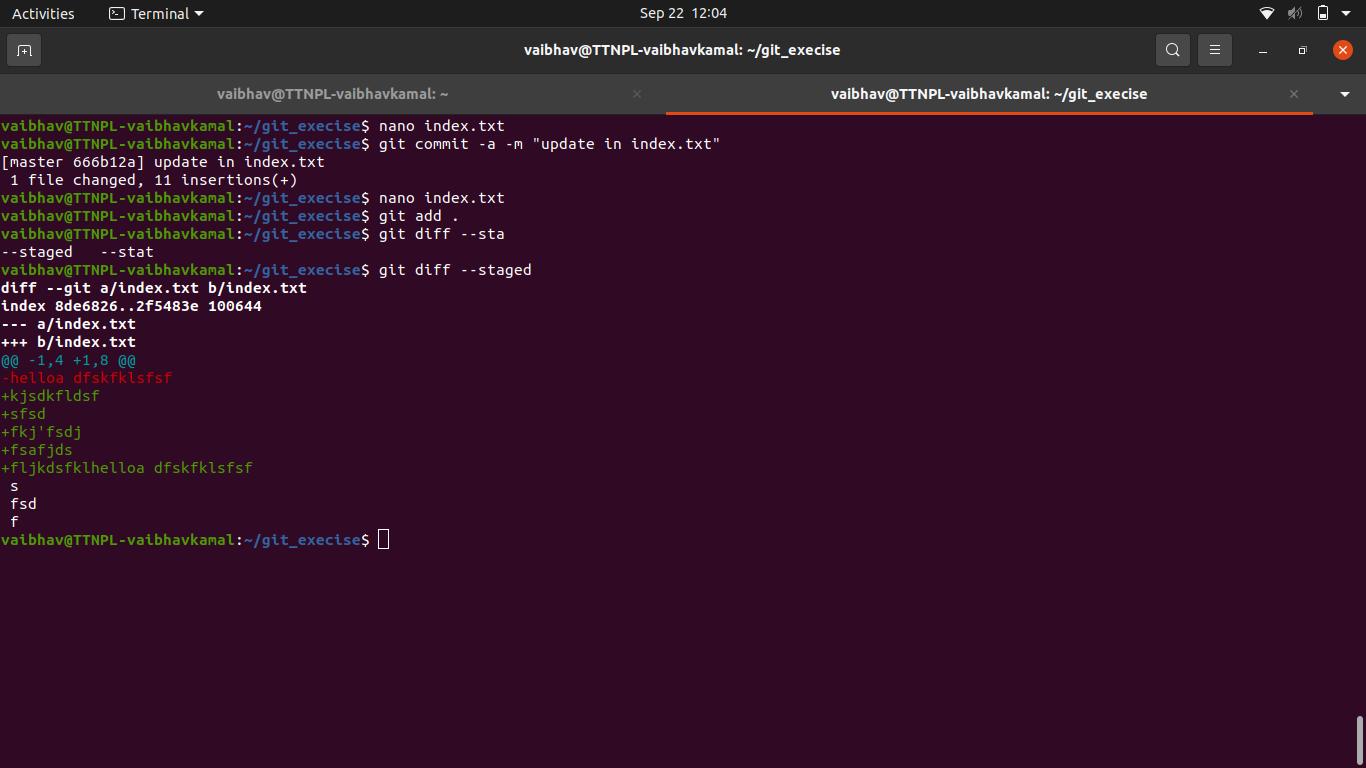
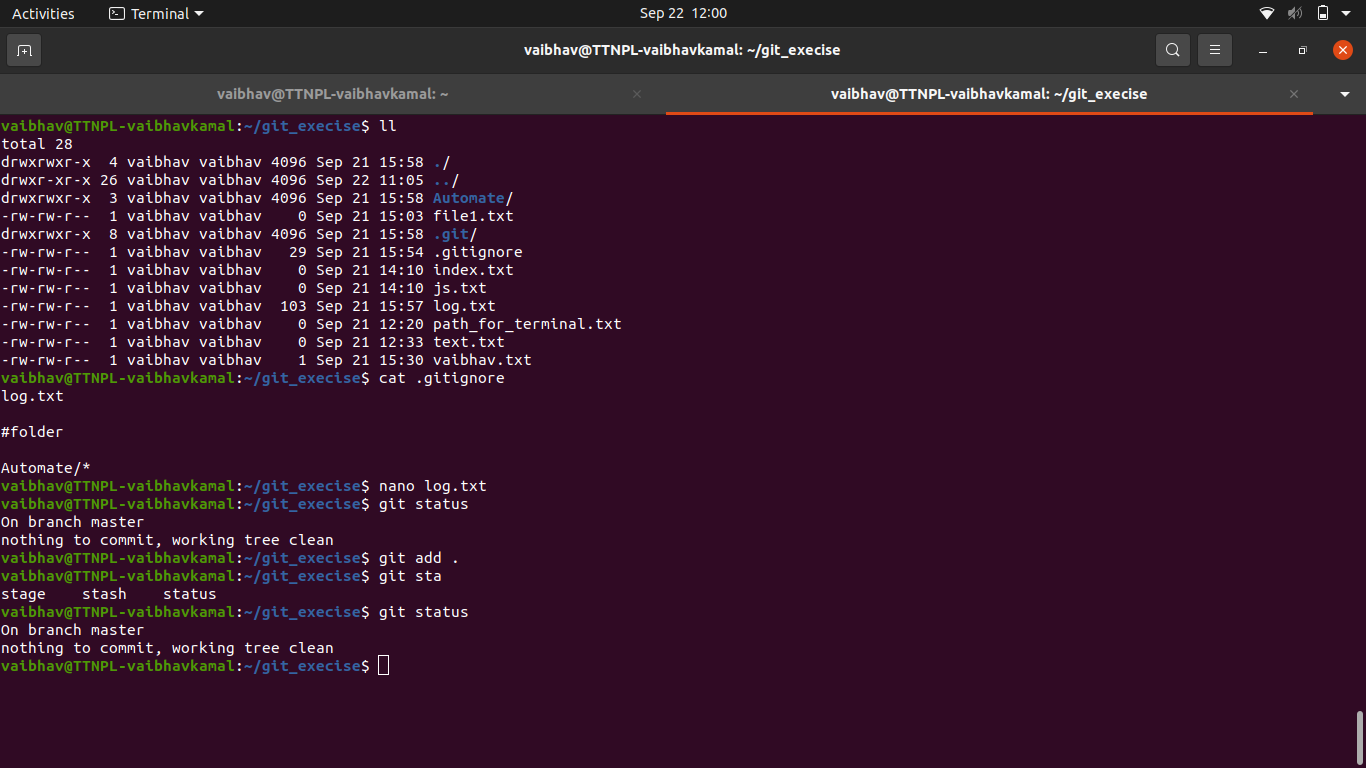
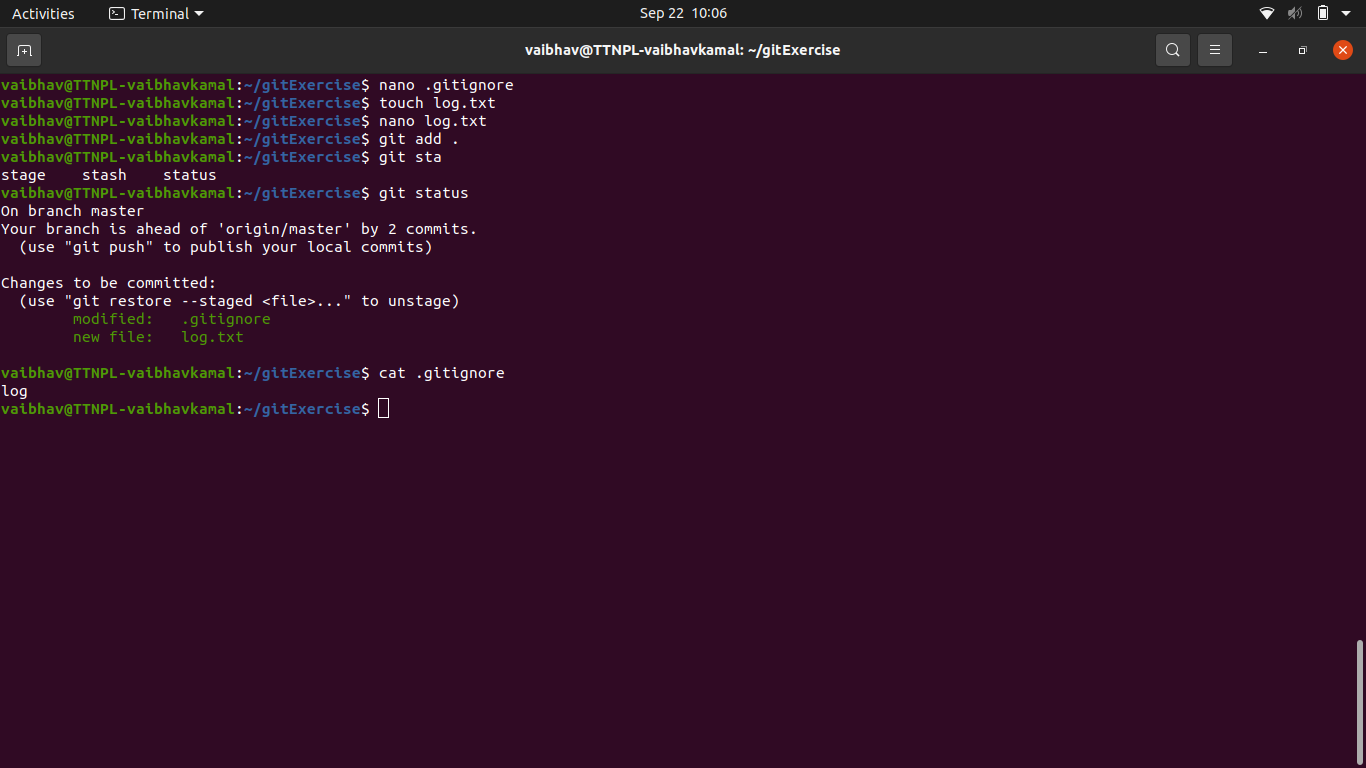
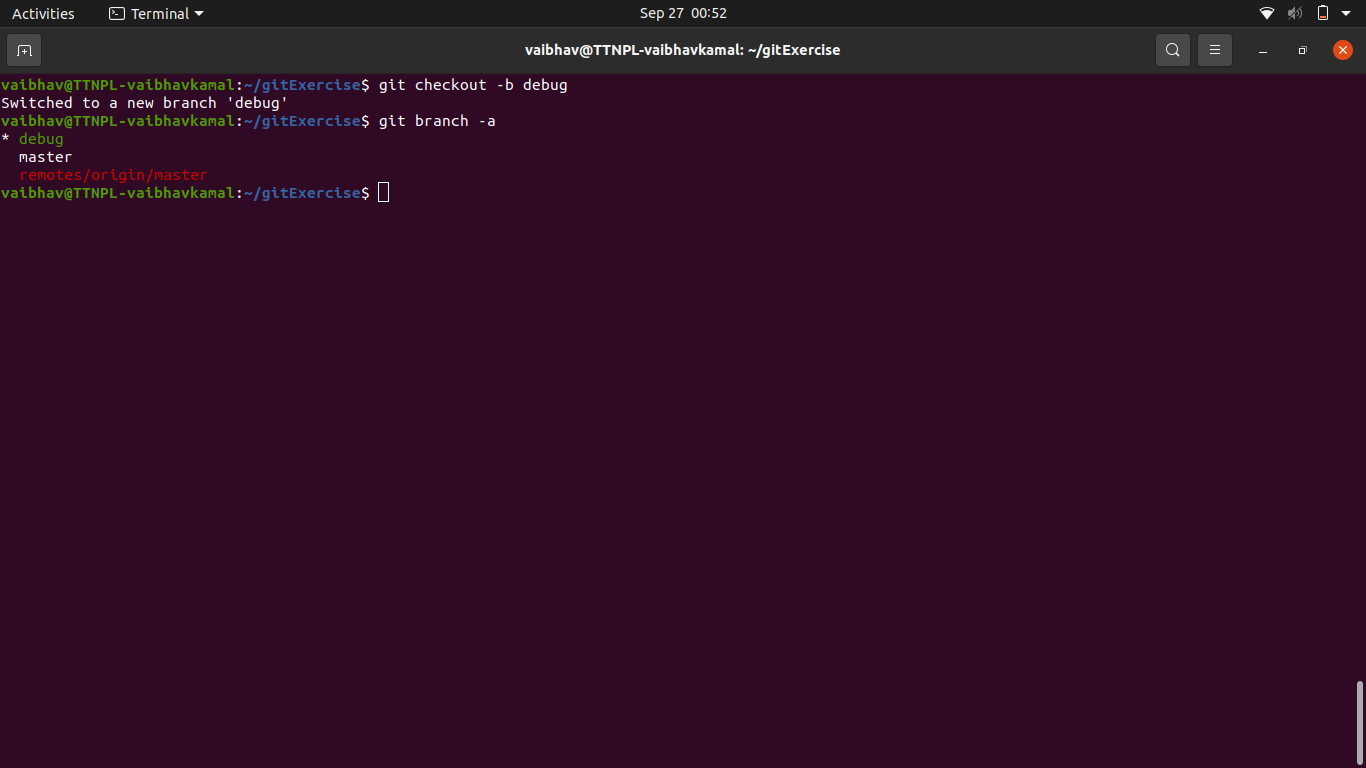
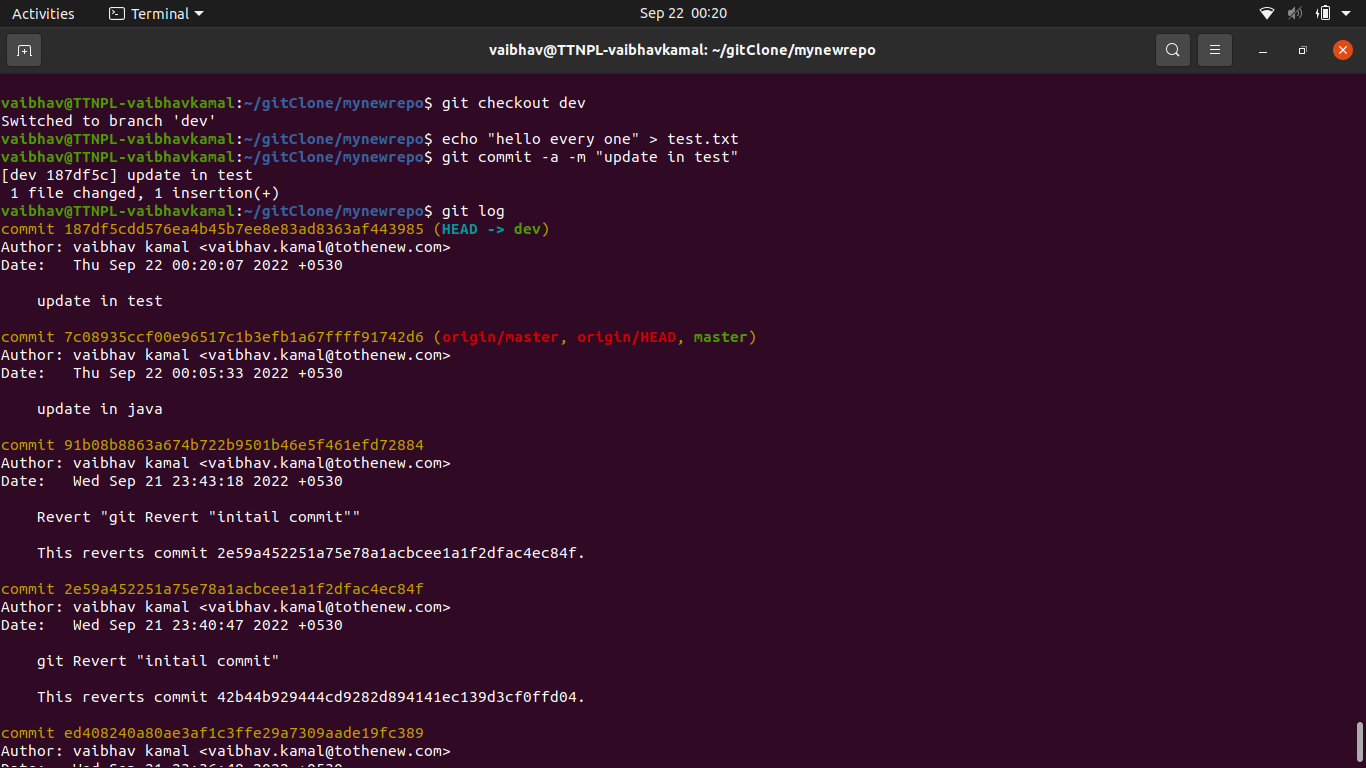
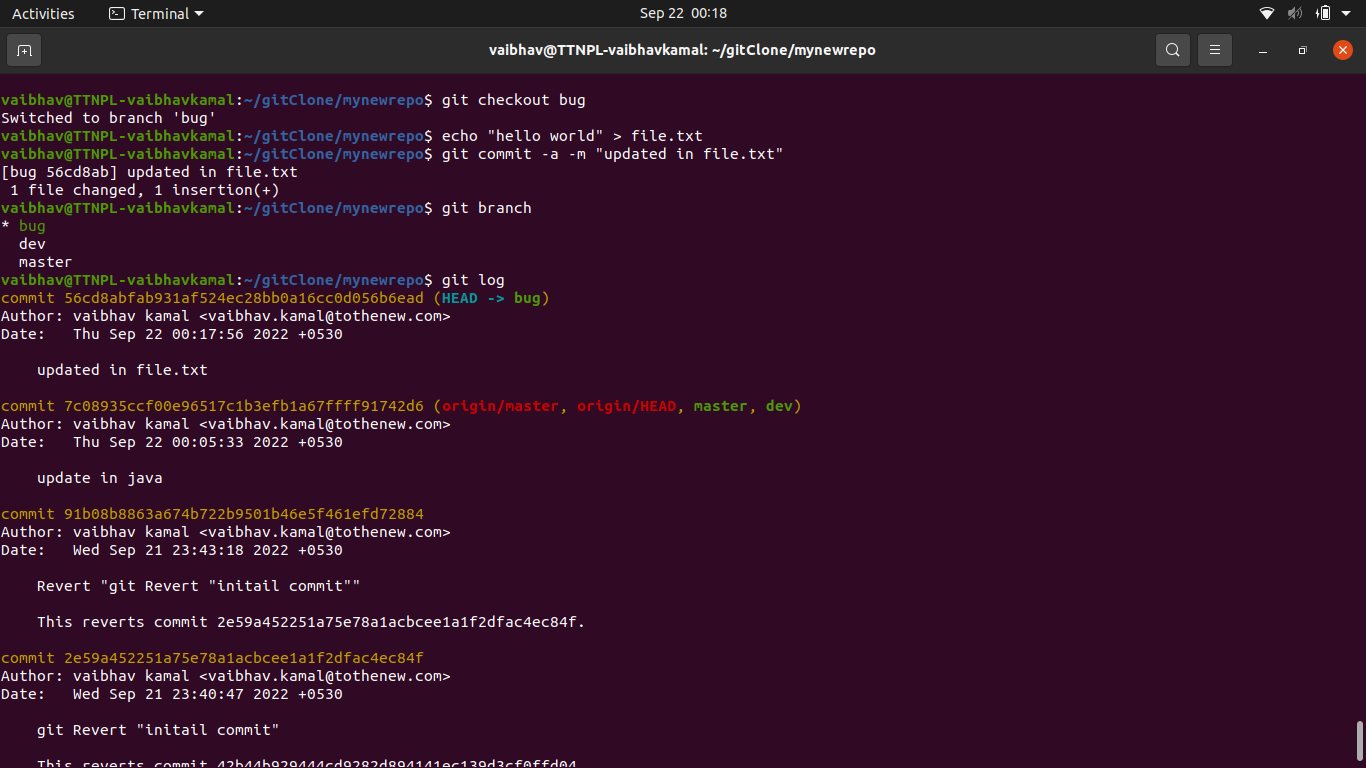
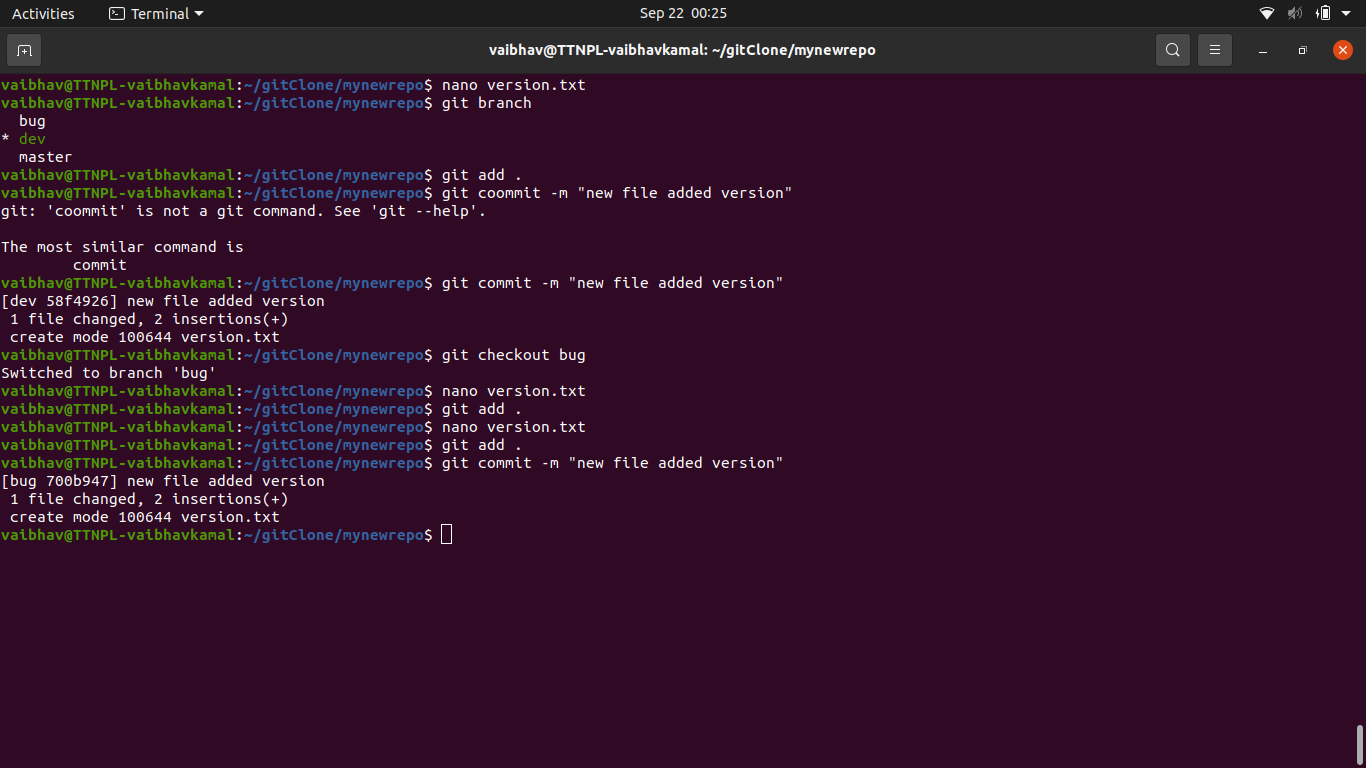
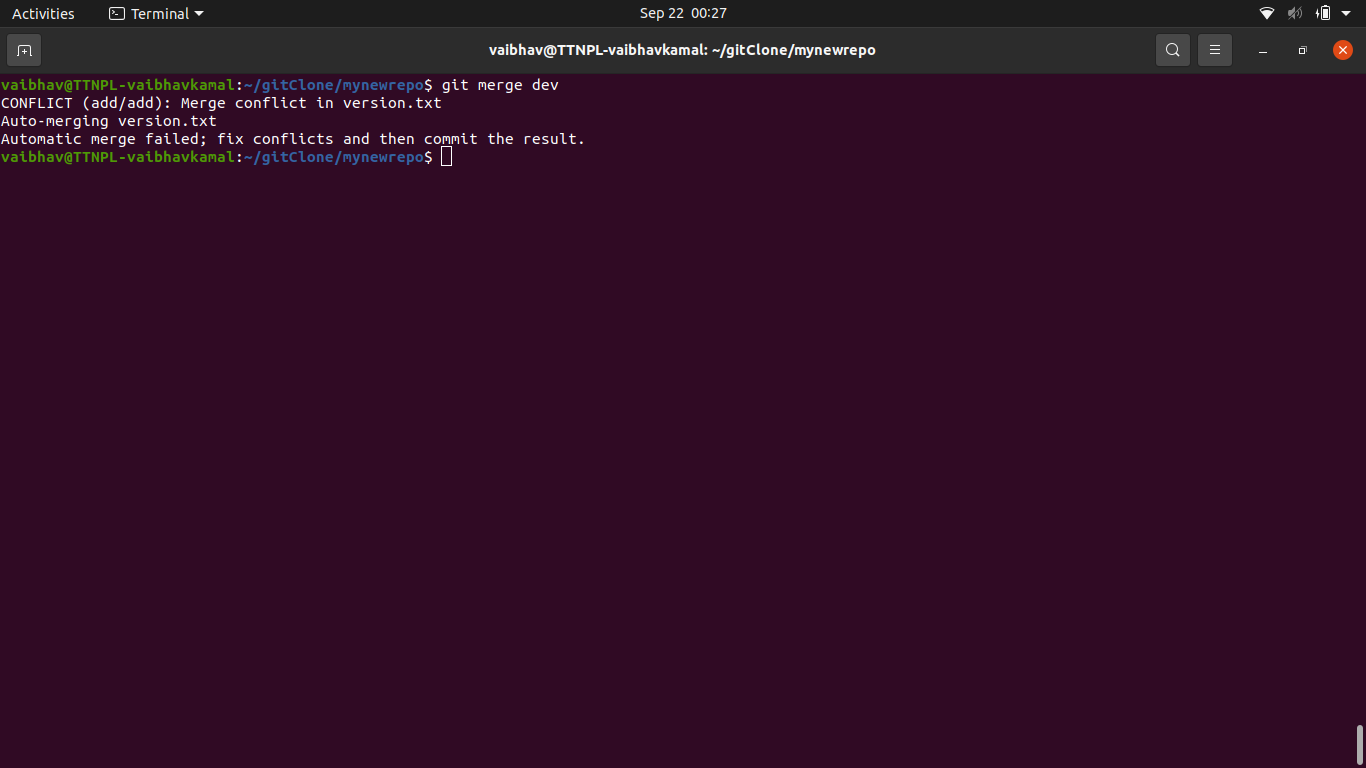
**Version Control Exercise**

1. **Git Setup** [**https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html**](https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html)
   1. **sudo apt-get install git**
   2. **git config --global user.name "vaibhav kamal"**
   3. **git config --global user.email "vaibhav.kamal@tothenew.com"**
2. **Initialize a Git Repository**
   1. **git init**
   2. ****
3. **Add files to the repository**
   1. **touch text.txt index.html test.txt file.txt js.txt css.txt**

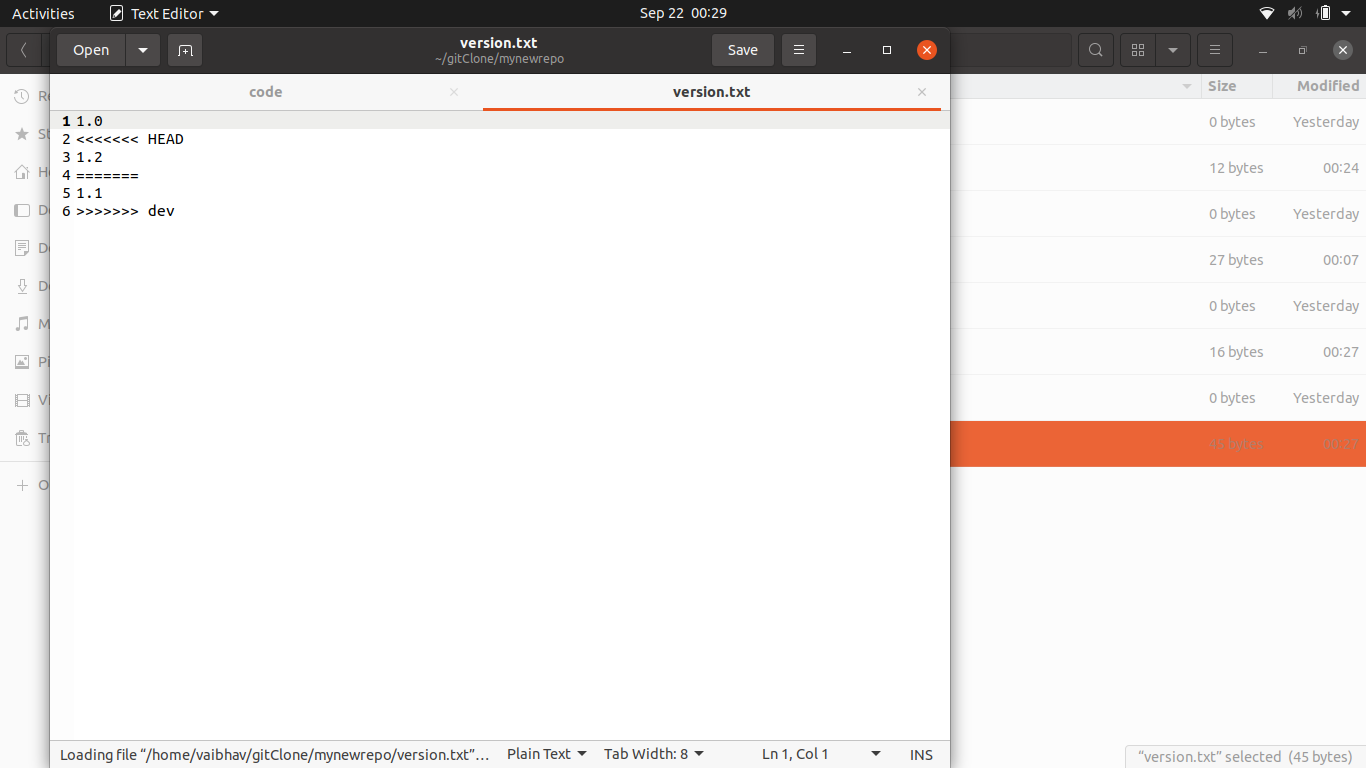
****

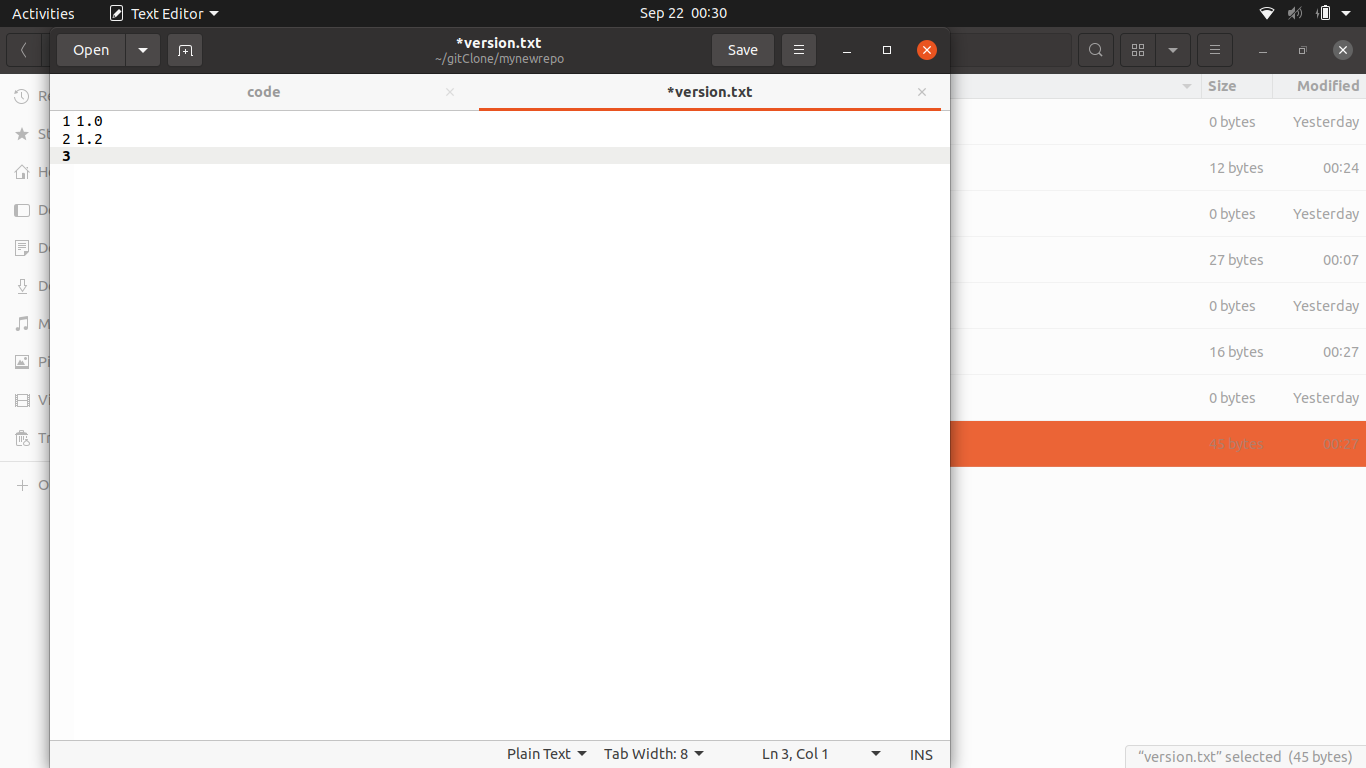
1. **Unstage 1 file**
   1. **git rm -cached file.txt**

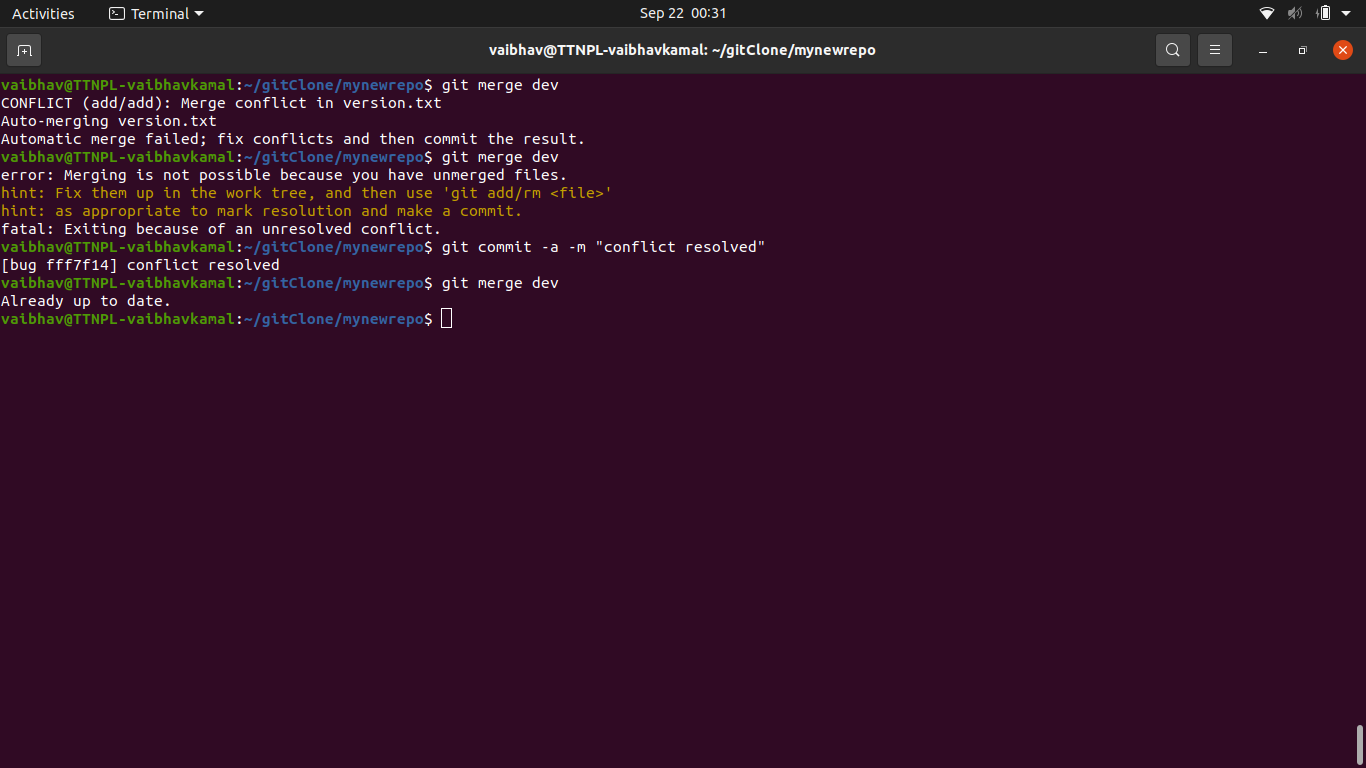
****

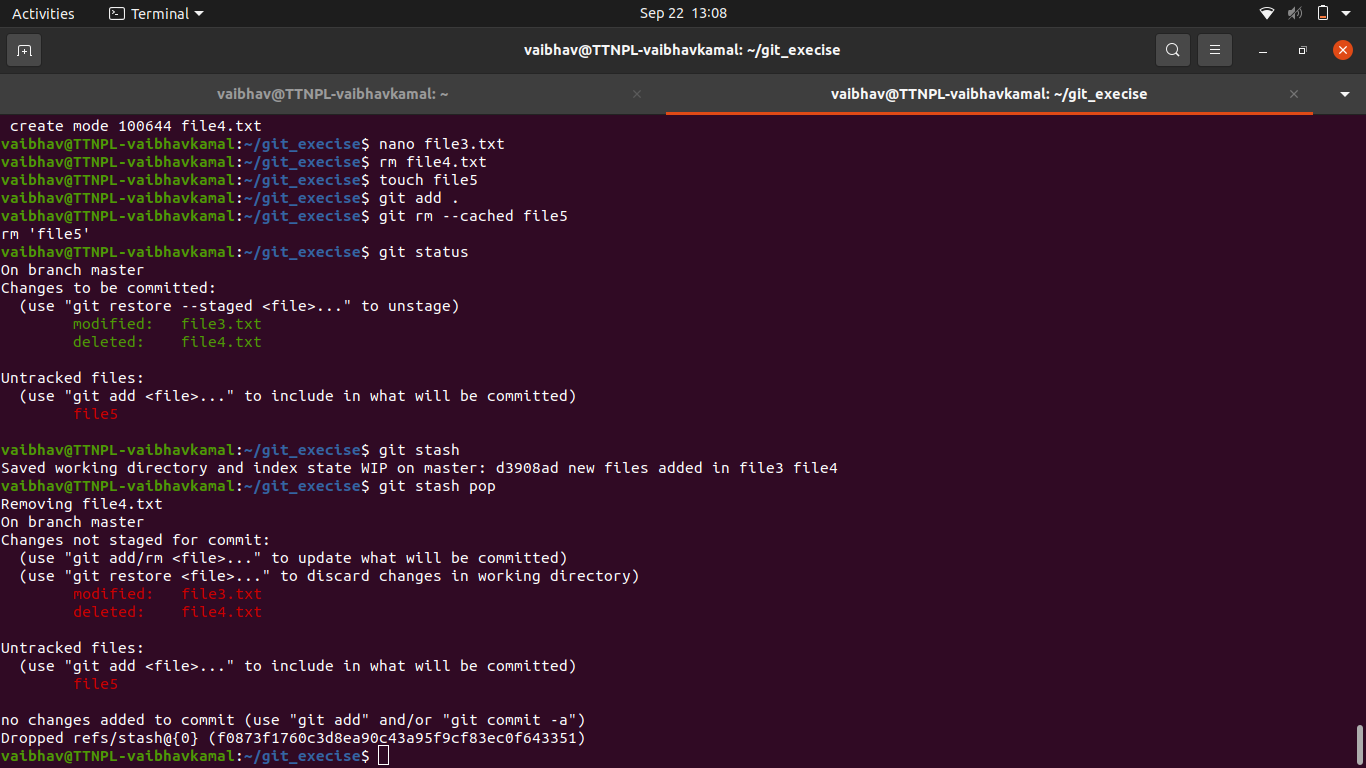
1. **Commit the file**
   1. **git commit -m “initail commit”**
   2. ****
2. **Add a remote**
   1. **git remote add origin** [**https://github.com/vaibhavtothenew/mynewrepo.git**](https://github.com/vaibhavtothenew/mynewrepo.git)
   2. ****
3. **Undo changes to a particular file**
   1. **git checkout file\_name**
   2. ****
4. **Push changes to Github**
   1. ****
5. **Clone the repository**
   1. ****
6. **Add changes to one of the copies and pull the changes in the other.**
   1. ****
   2. ****
7. **Check differences between a file and its staged version**
   1. ****
8. **Ignore a few files to be checked in**
   1. ****
   2. ****
9. **Create a new branch.**
   1. ****
10. **Diverge them with commits**
    1. ****
11. **Edit the same file at the same line on both branches and commit**
    1. ****
12. **Try merging and resolve merge conflicts**
    1. ****

**Manually make change on the line we want to keep the code**

****

****

****

1. **Stash the changes and pop them**
   1. ****
2. **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**
   1. 