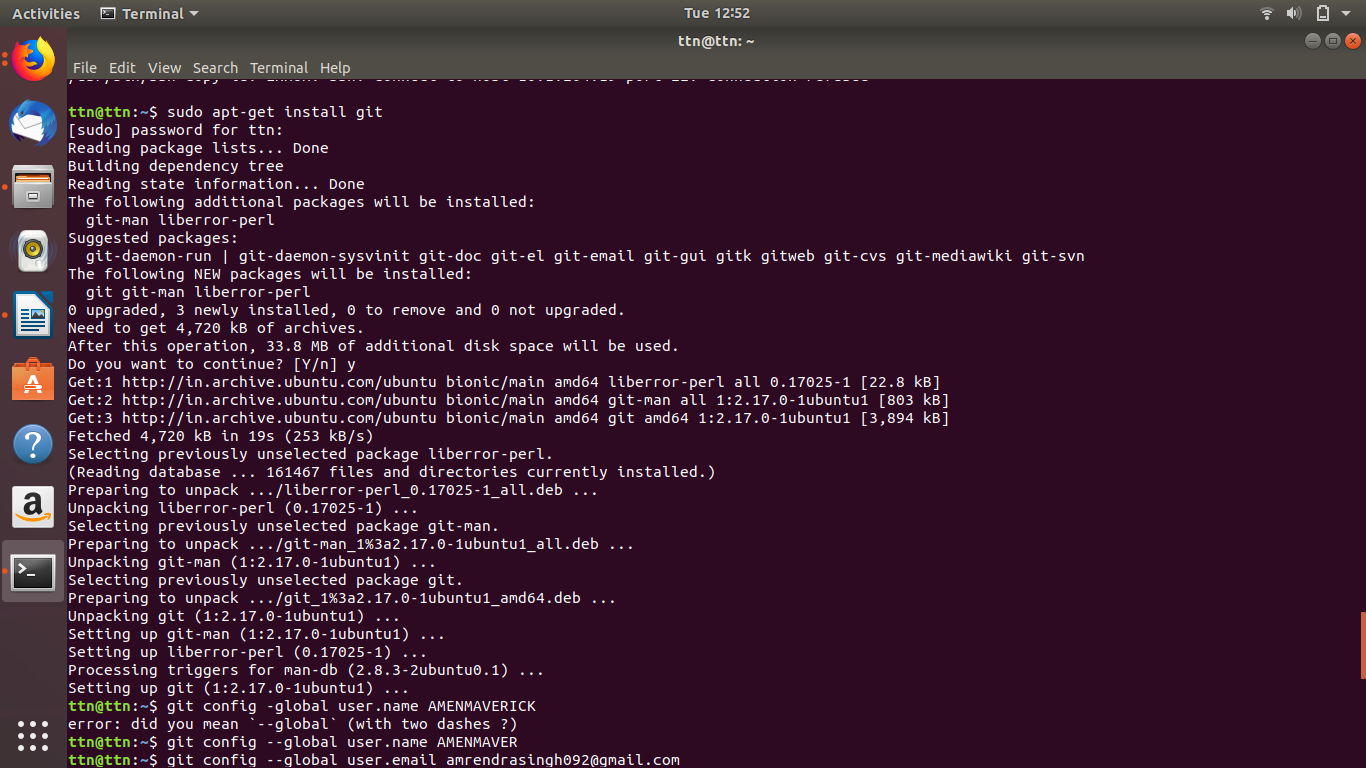
**ASSIGNMENT - 2**

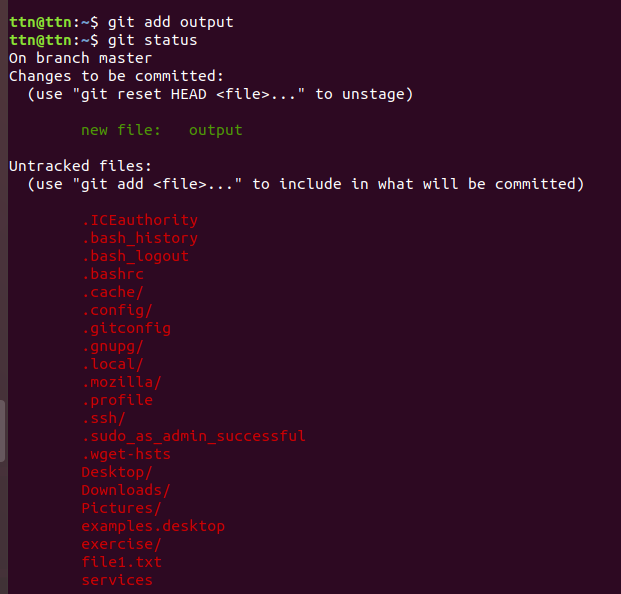
**Q.1.)** Git Setup.



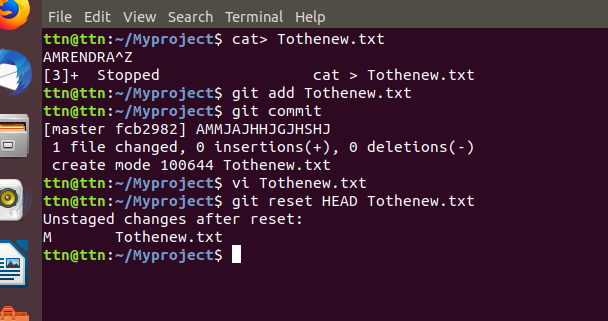
**Q.2.)** Initialize a Git Repository



**Q.3.)** Add files to the repository.

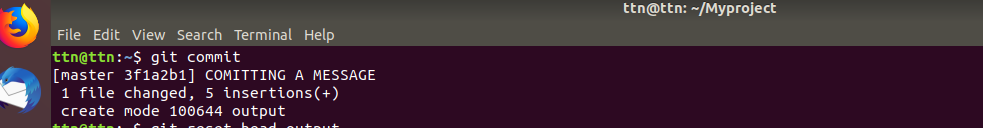


**Q.4.)** Unstage 1 file



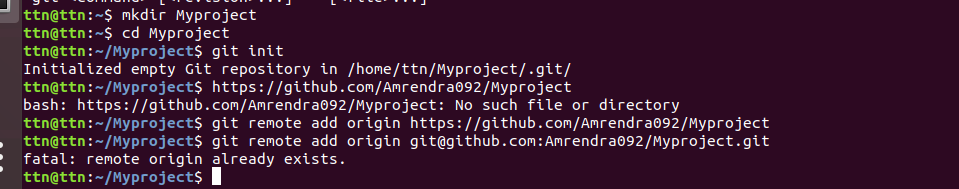
**Q.5.) Commit the file.**

**Solution :**

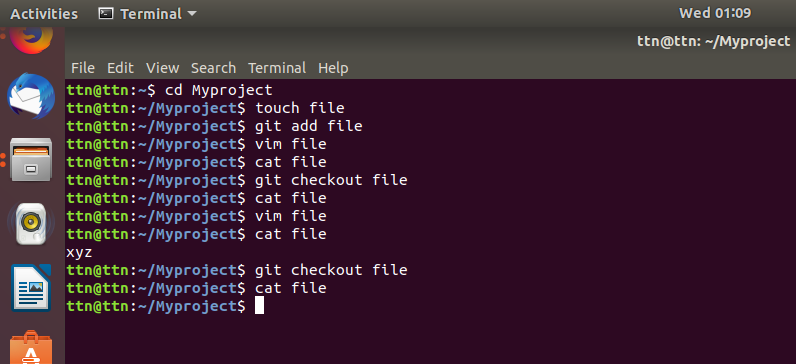


**Q.6.) Add a remote.**

**Solution :**

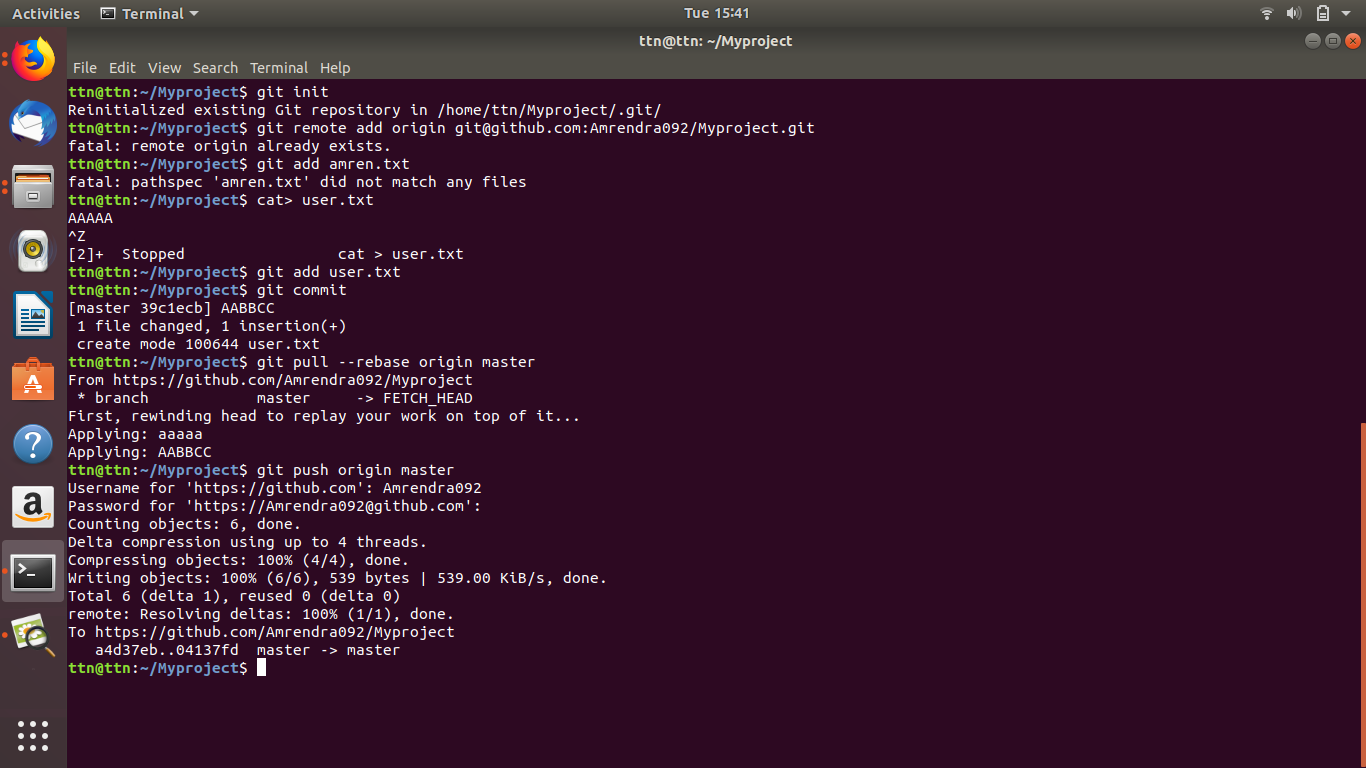


**Q.7.) Undo changes to a particular file.**

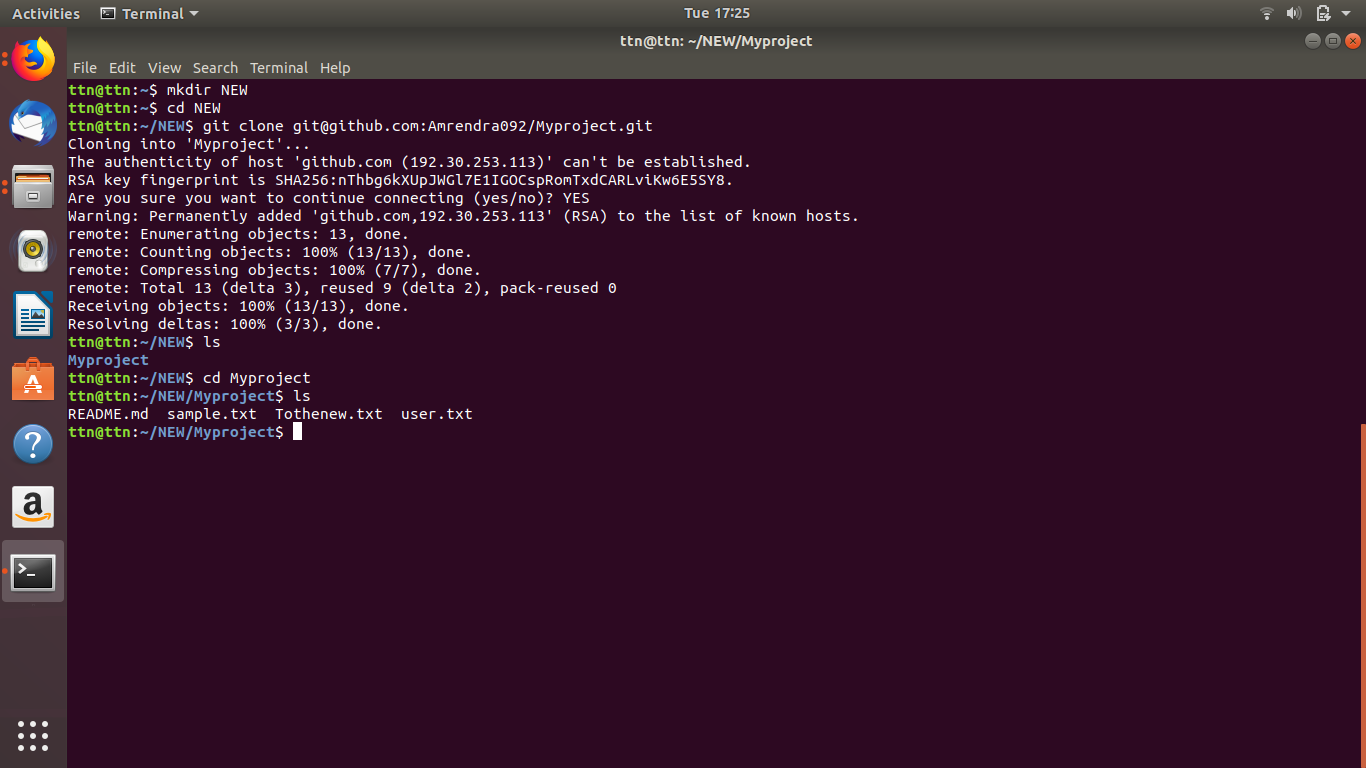
****

**Q.8.) Push changes to Github.**

**Solution :**

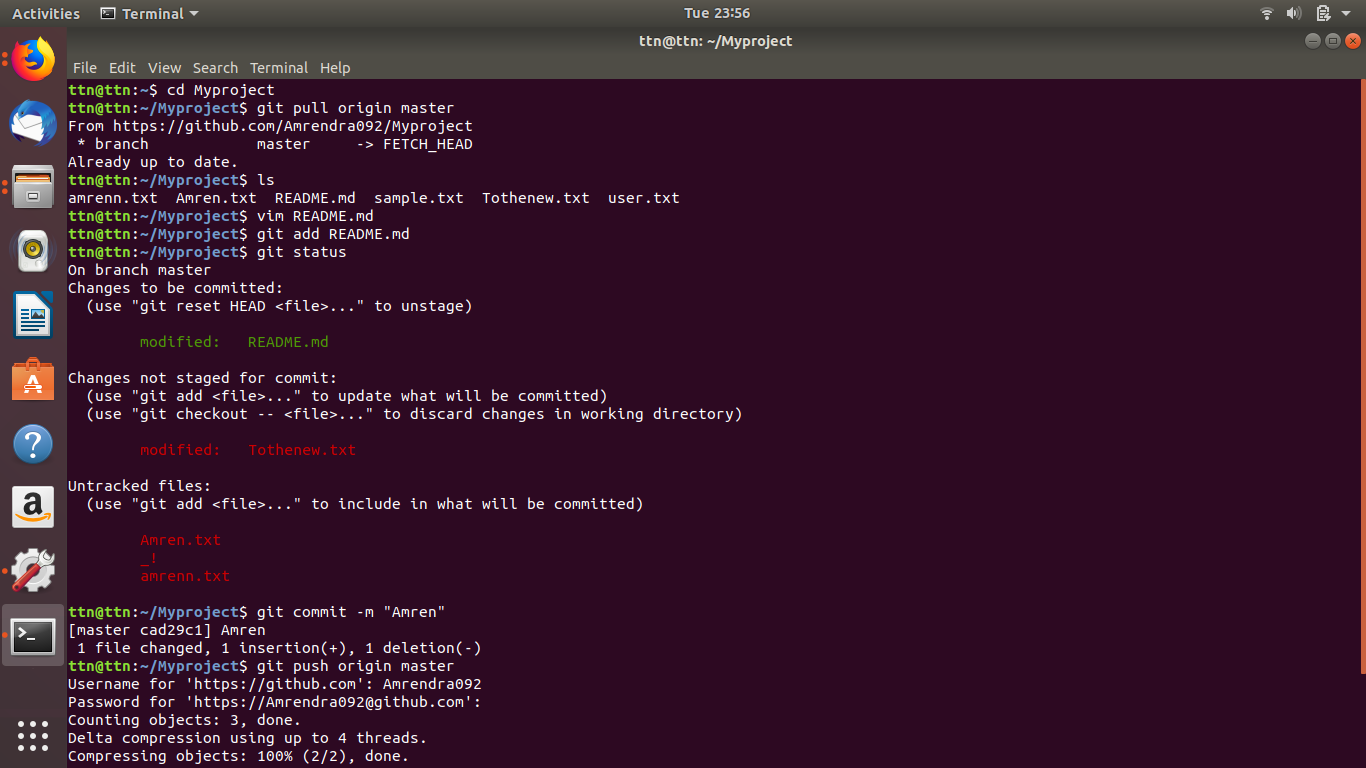


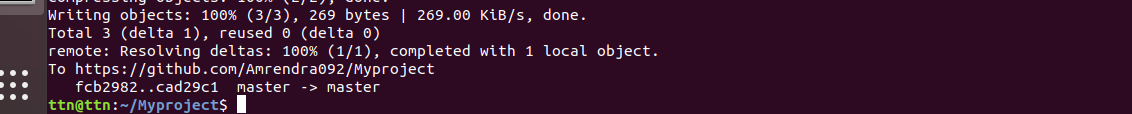
**Q.9.) Clone the repository.**

**Solution :**

**Q.10.) Add changes to one of the copies and pull the changes in the other.**

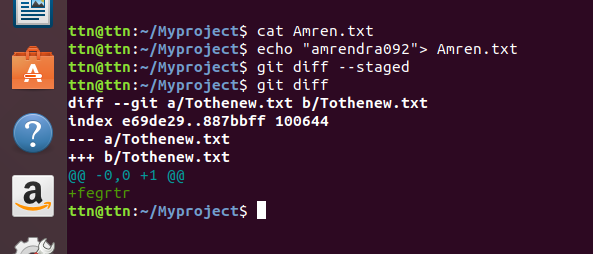
**Solution :**

****

****

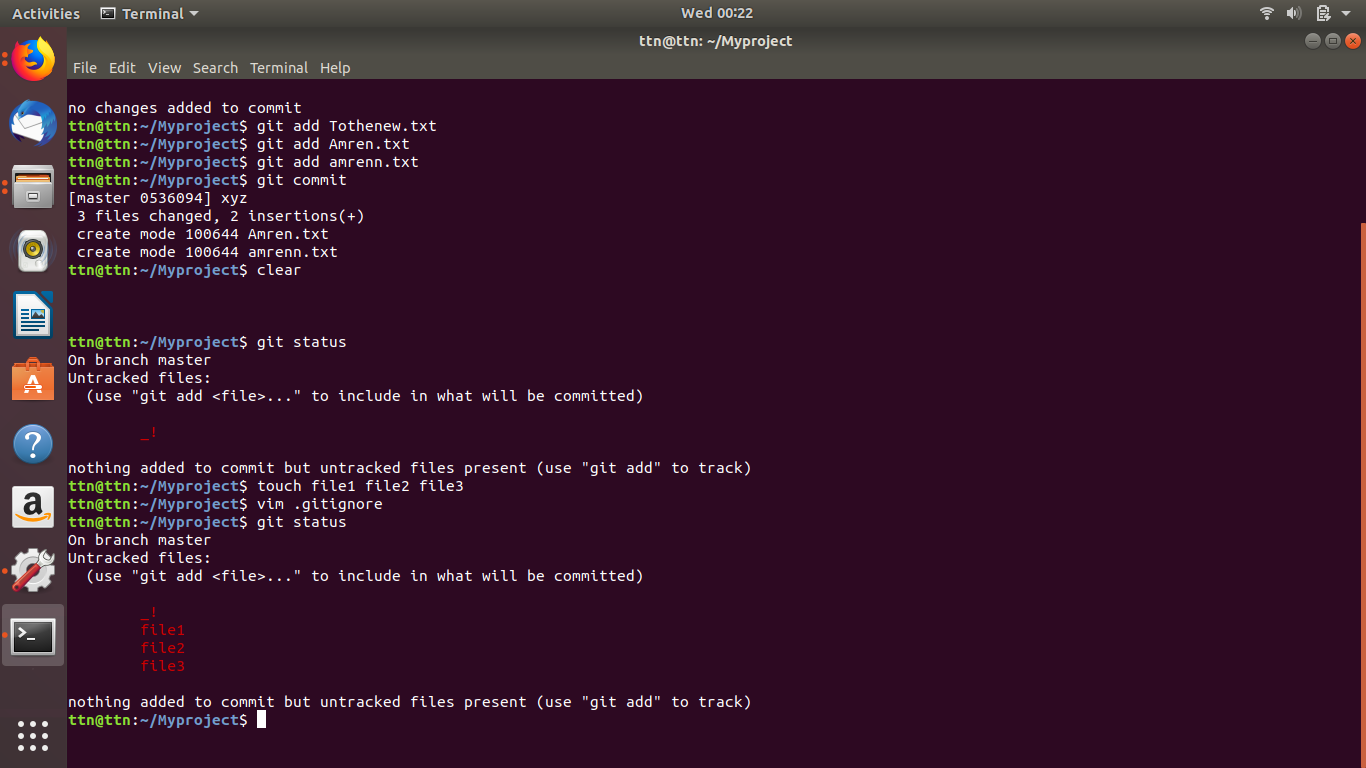
**Q.11.) Check differences between a file and its staged version.**

**Solution :**

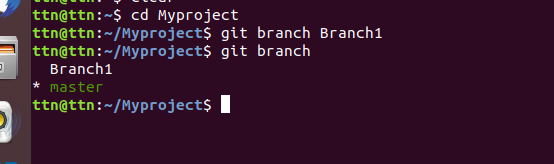
****

**Q. 12 ) Ignore a few files to be checked in.**

**Solution :**

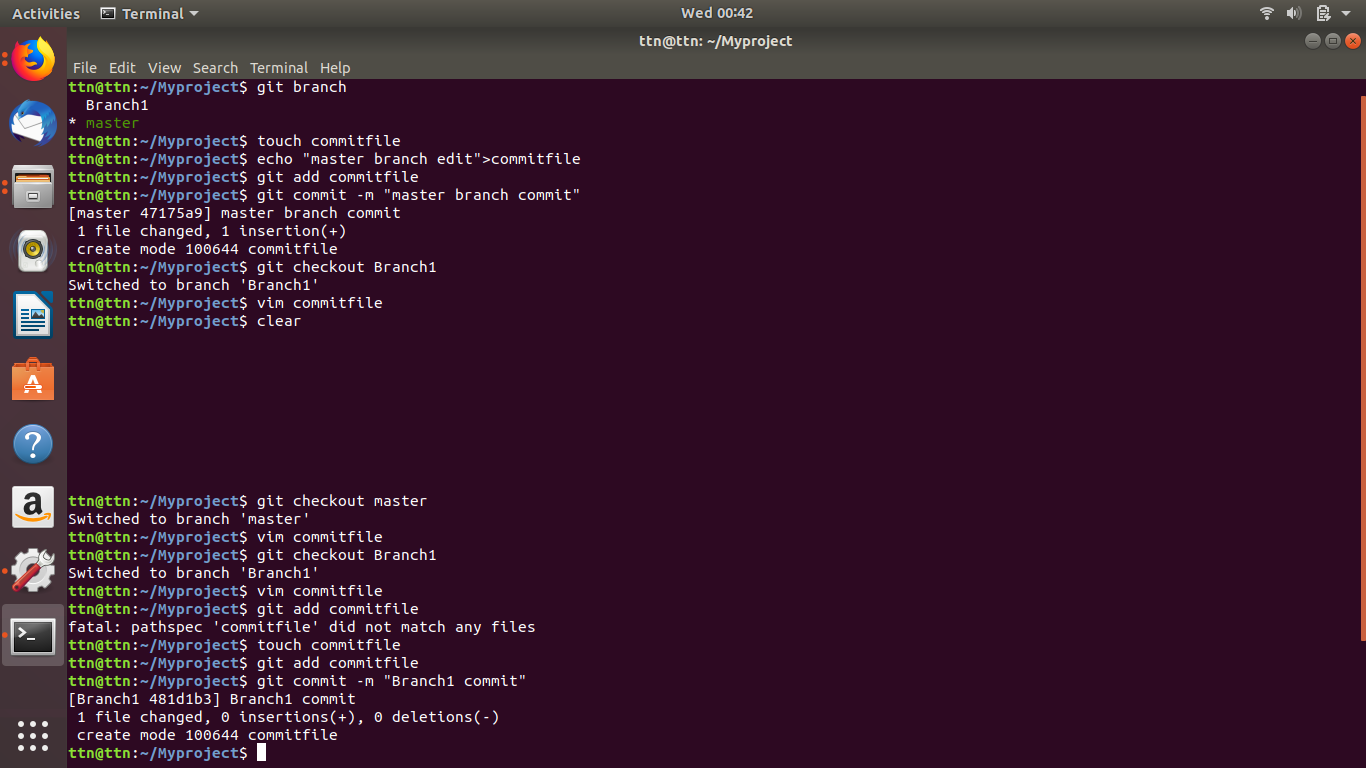
****

**Q.13 ) Create a new branch.**

**Solution :**

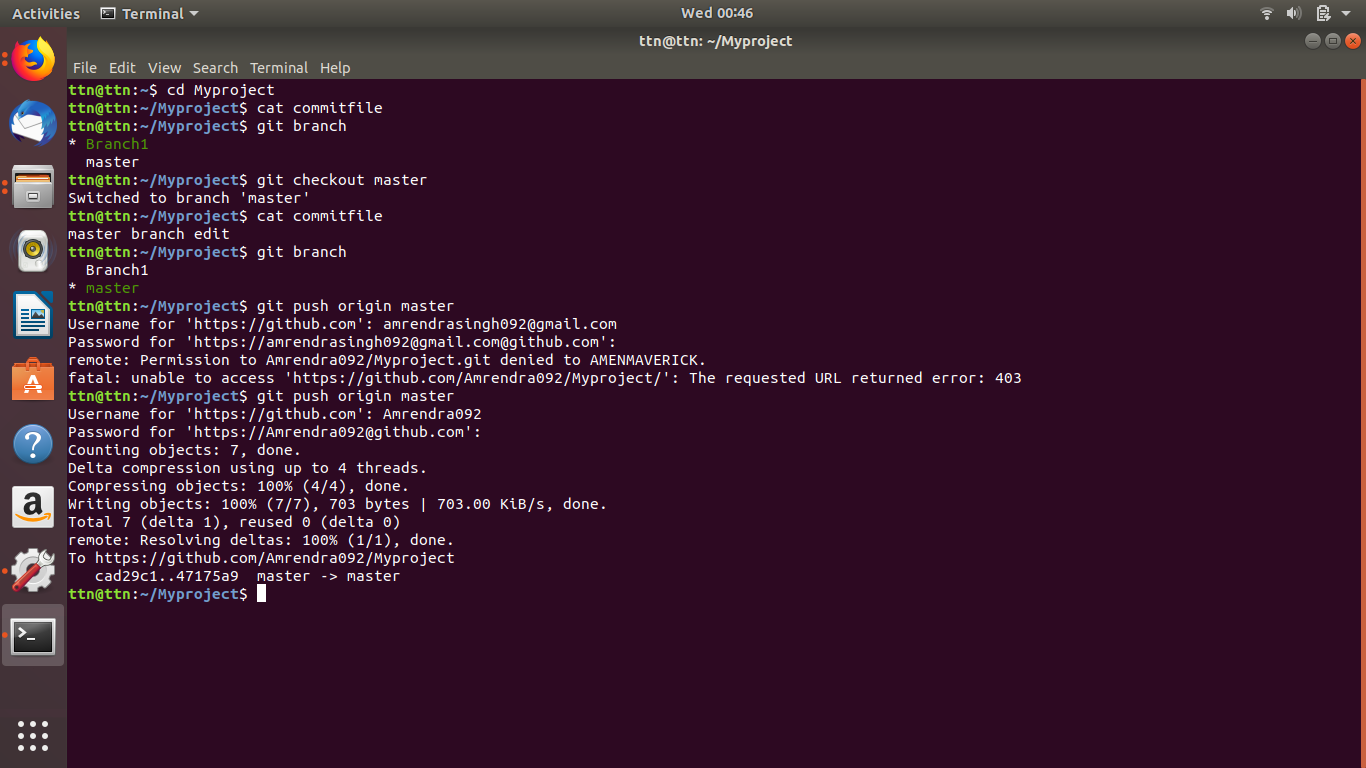
**Q.14. ) Diverge them with commits.**

**Solution :**

****

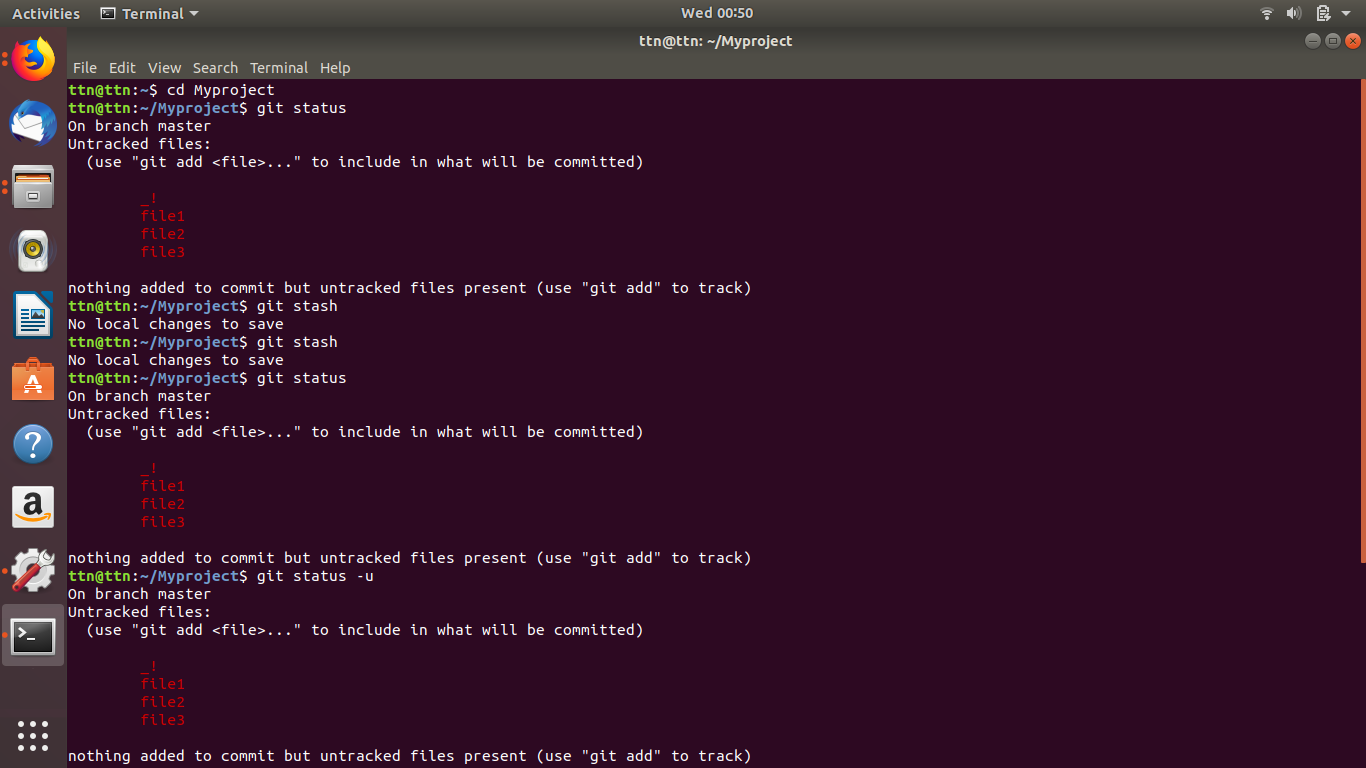
**Q. 15.) Edit the same file at the same line on both branches and commit.**

**Solution :**

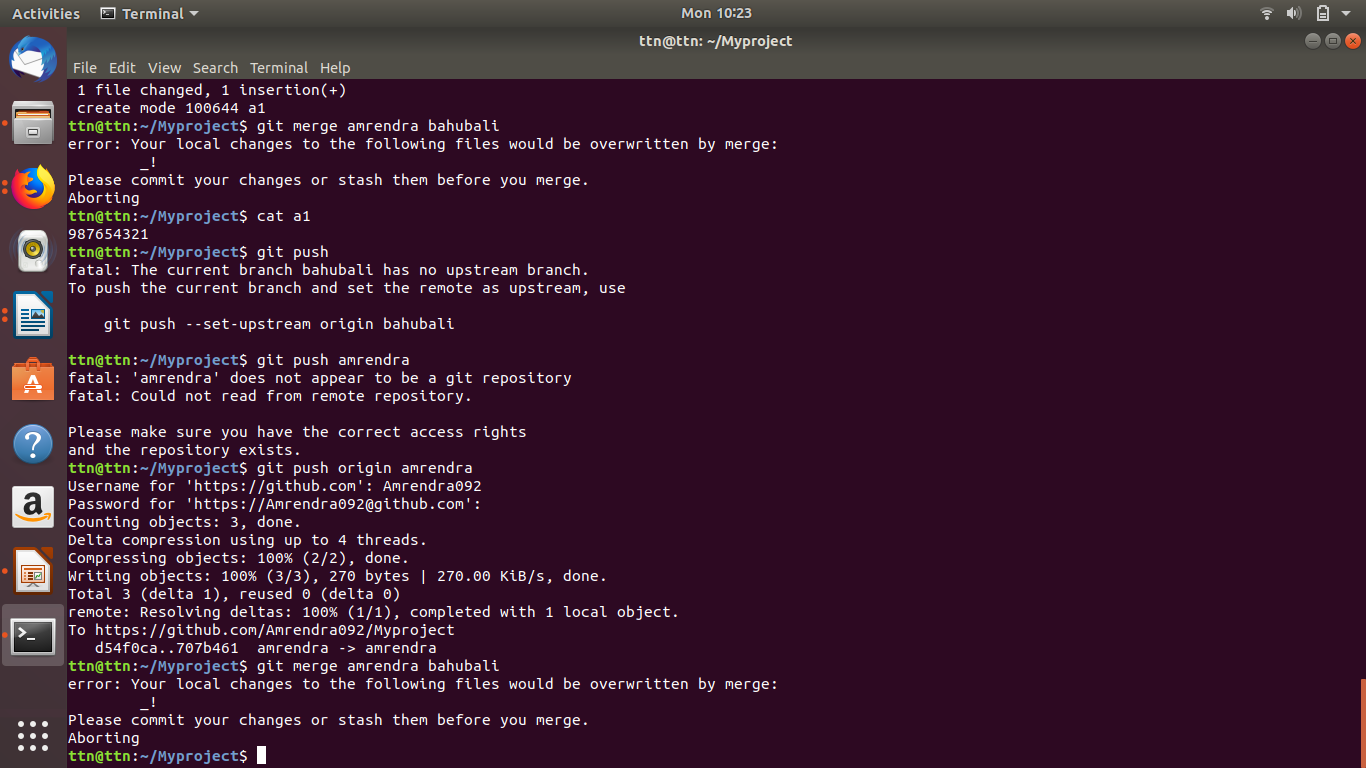
****

**Q.17 ) Stash the changes and pop them.**

**Solution :**



**Q.18) Try merging and resolve merge conflicts.**

**Solution :**

**Q.19 ) 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)\[\033[00m\]\$ '**

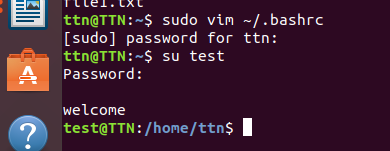
**else**

**PS1='\u@\h:\W $(parse\_git\_branch)\$ '**

**fi**

**unset color\_prompt force\_color\_prompt**

**Solution :**

****