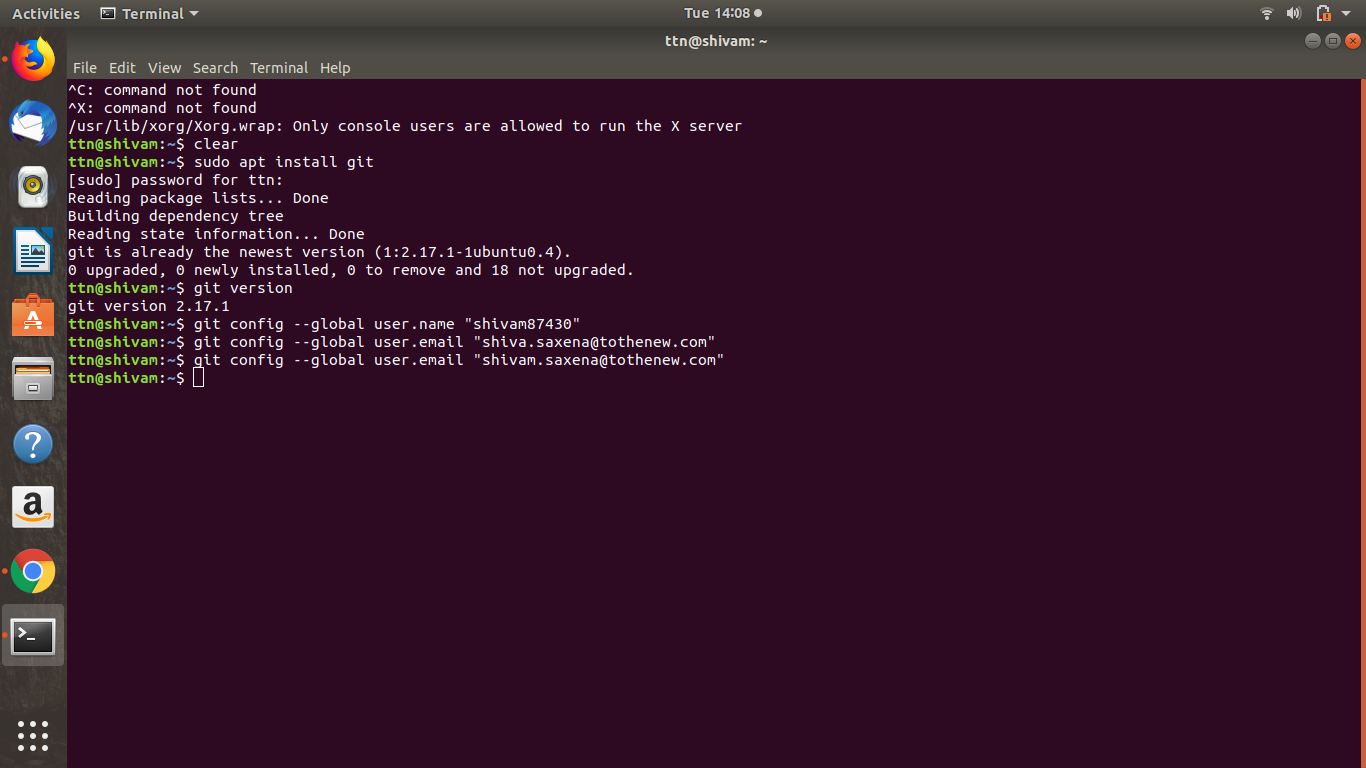
**Question 1 :** Git Setup https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html

**Answer 1 :**  sudo get install git

git version

git config --global user.name “username”

git config --global user.email “useremail”

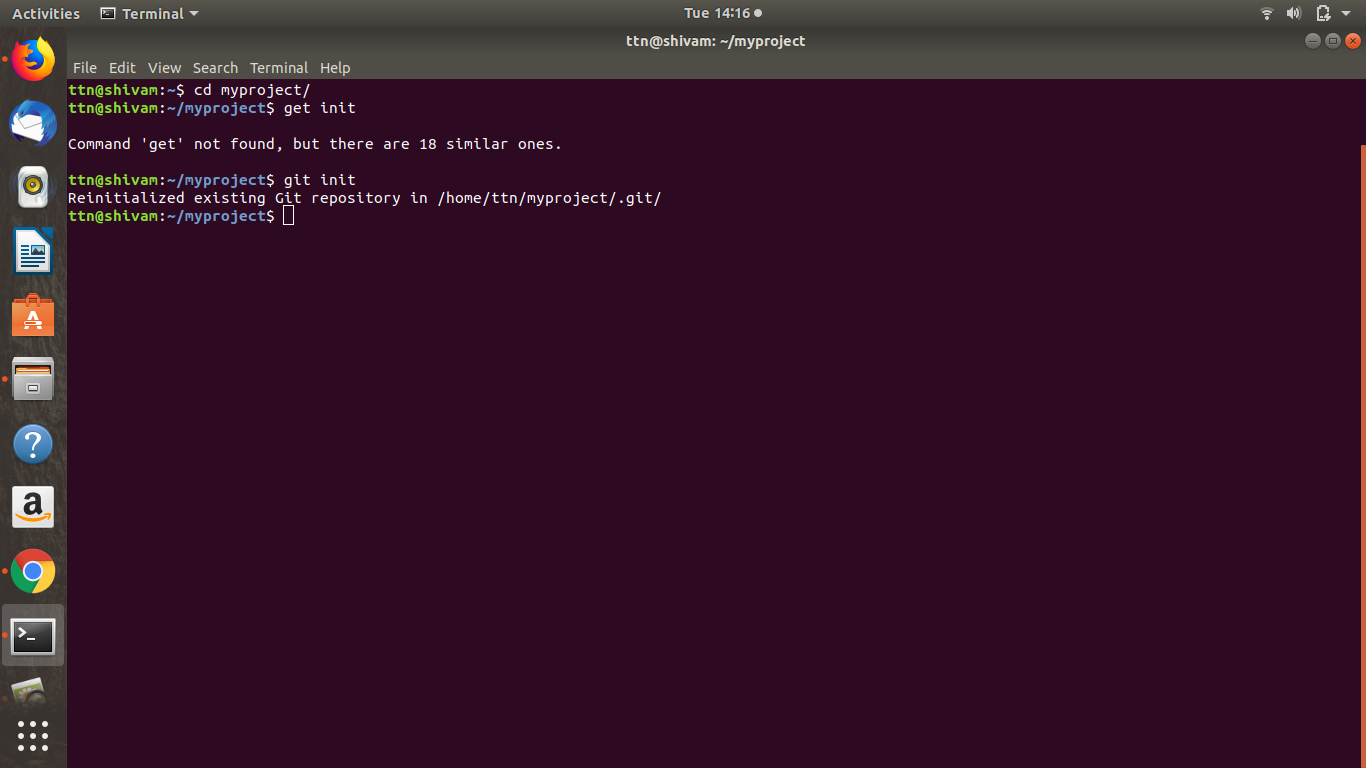


**Question 2 :** Initialize a Git Repository

**Answer 2 :**

Go to directory that we want to initialize

Command :- git init



**Question 3 :** Add files to the repository

**Answer 3 :**

Go to directory that we want to initialize

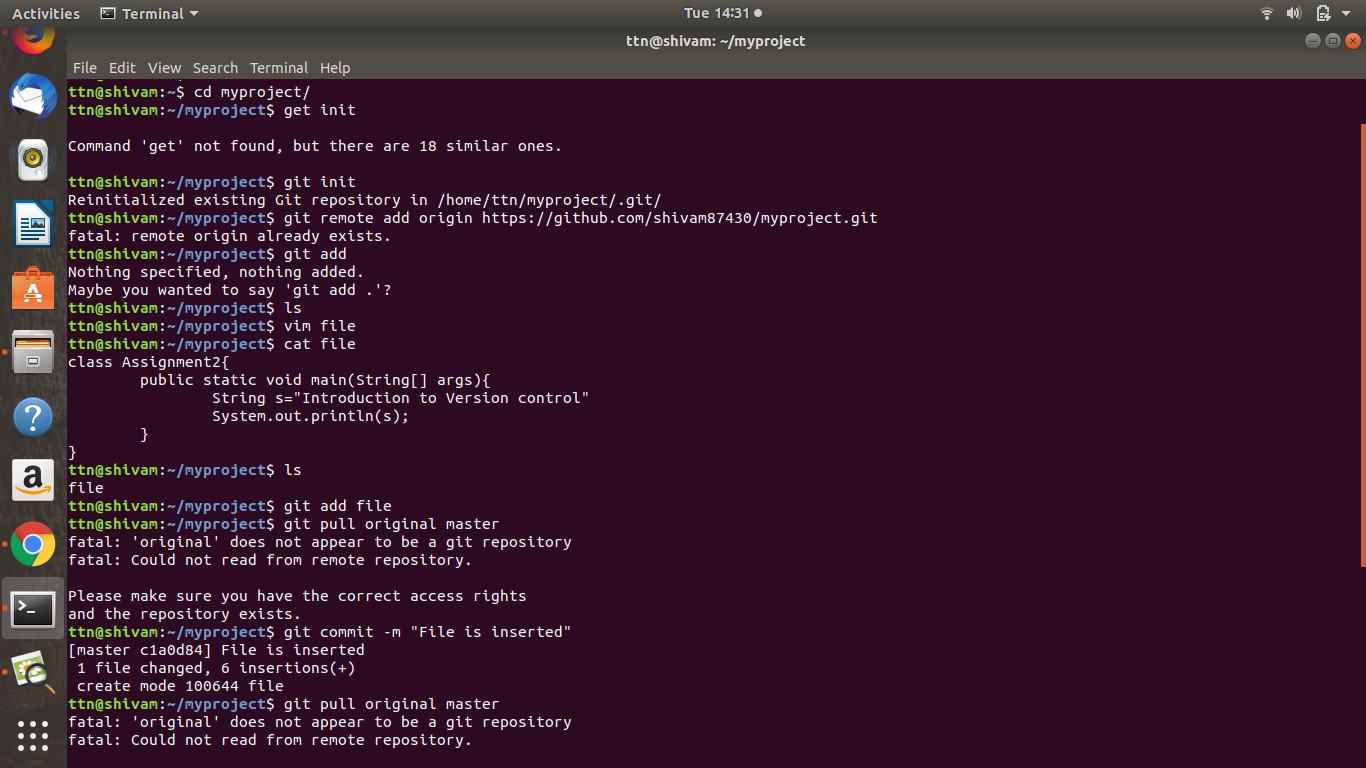
Create a file that we want to add through vmi

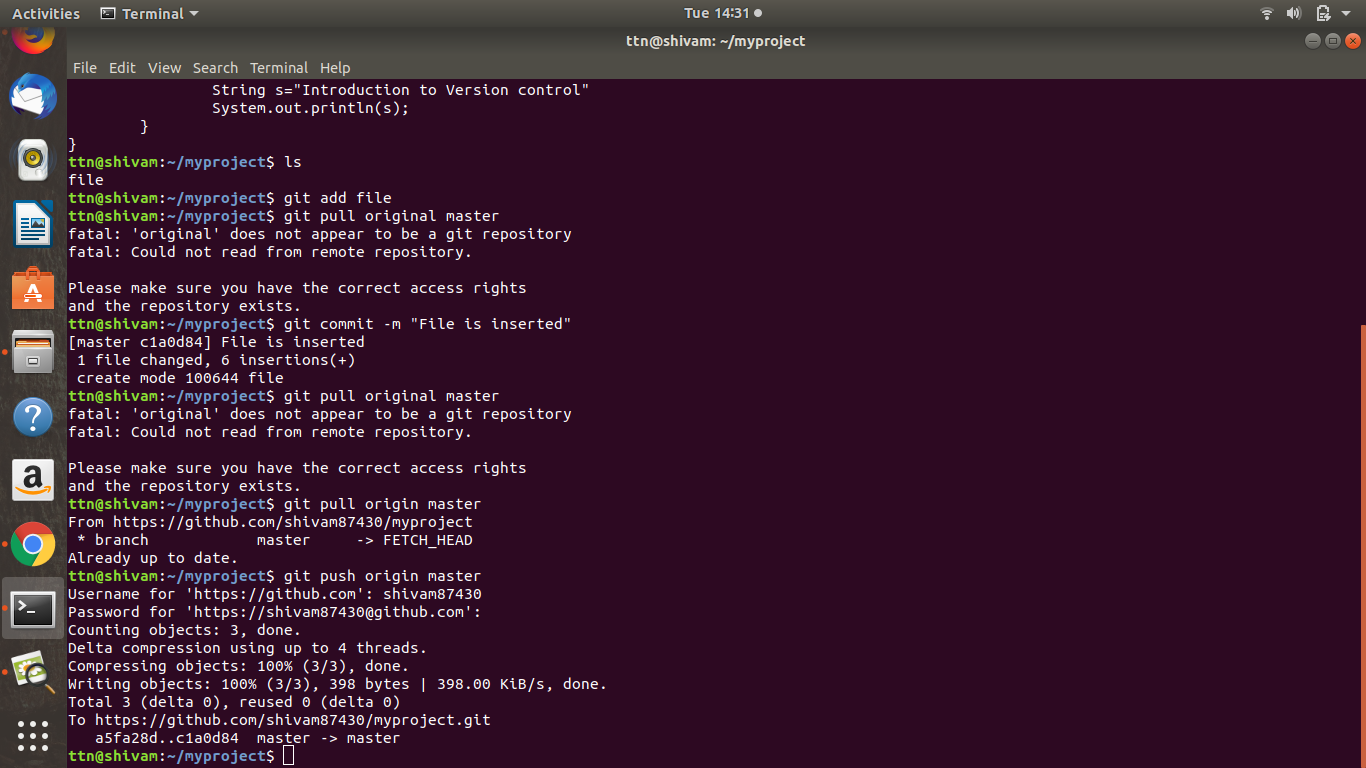
Command vmi file

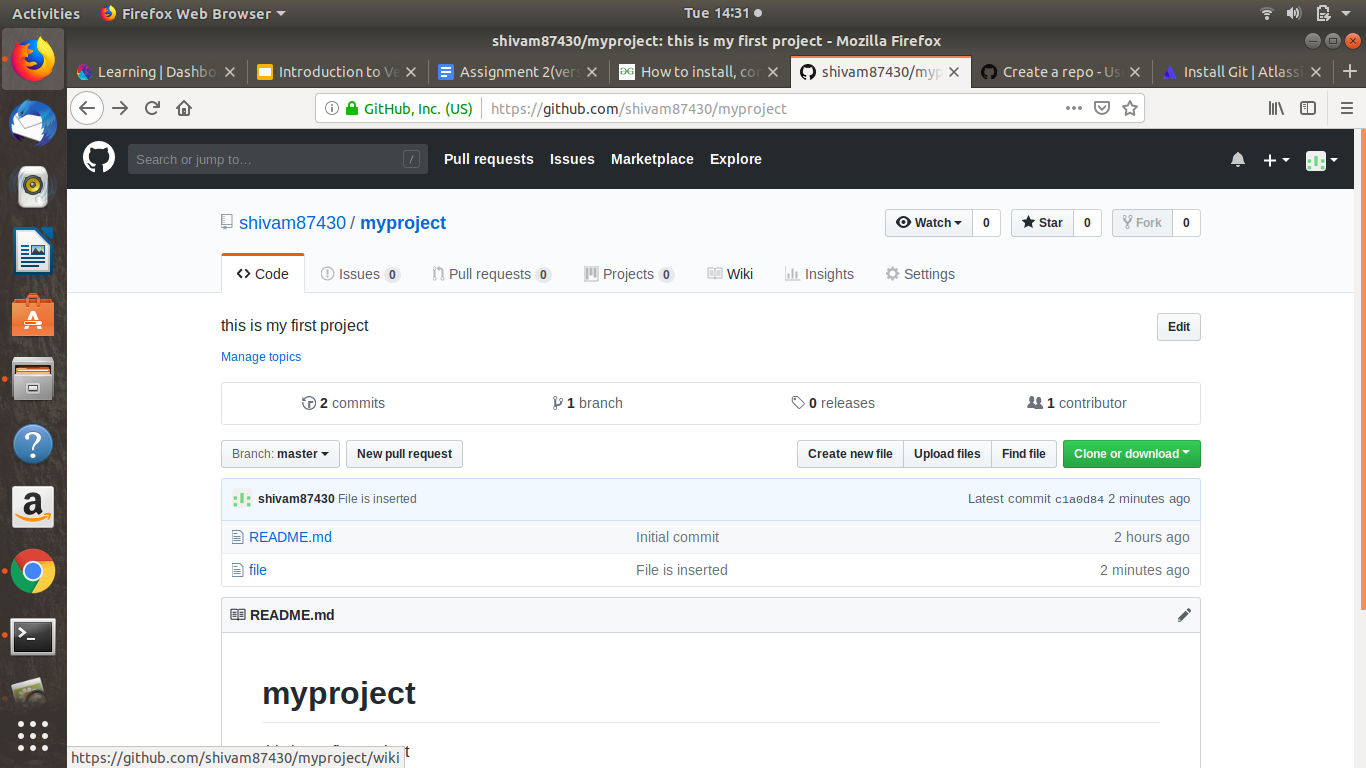
git add file

git commit -m “file is inserted”

git push origin master

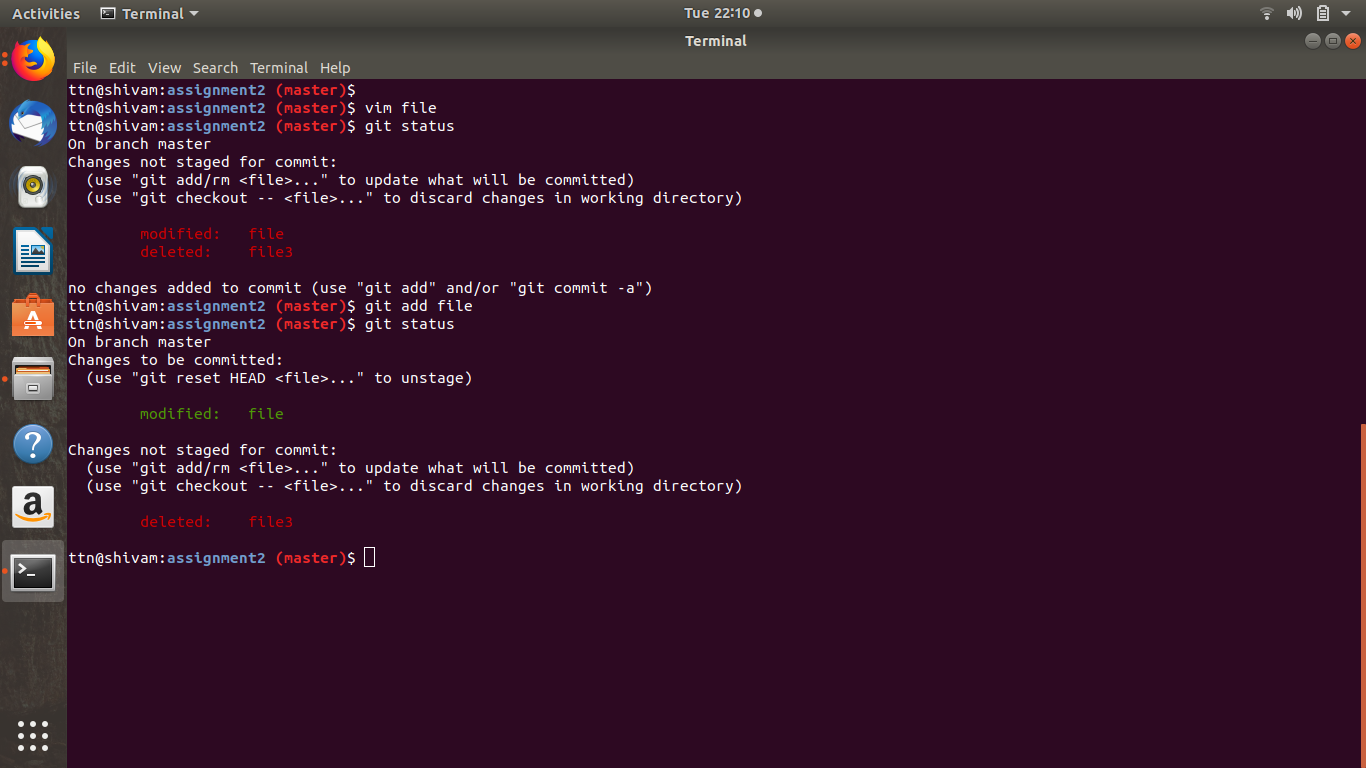






**Question 4 :** unstage 1 file.

**Answer 4.**



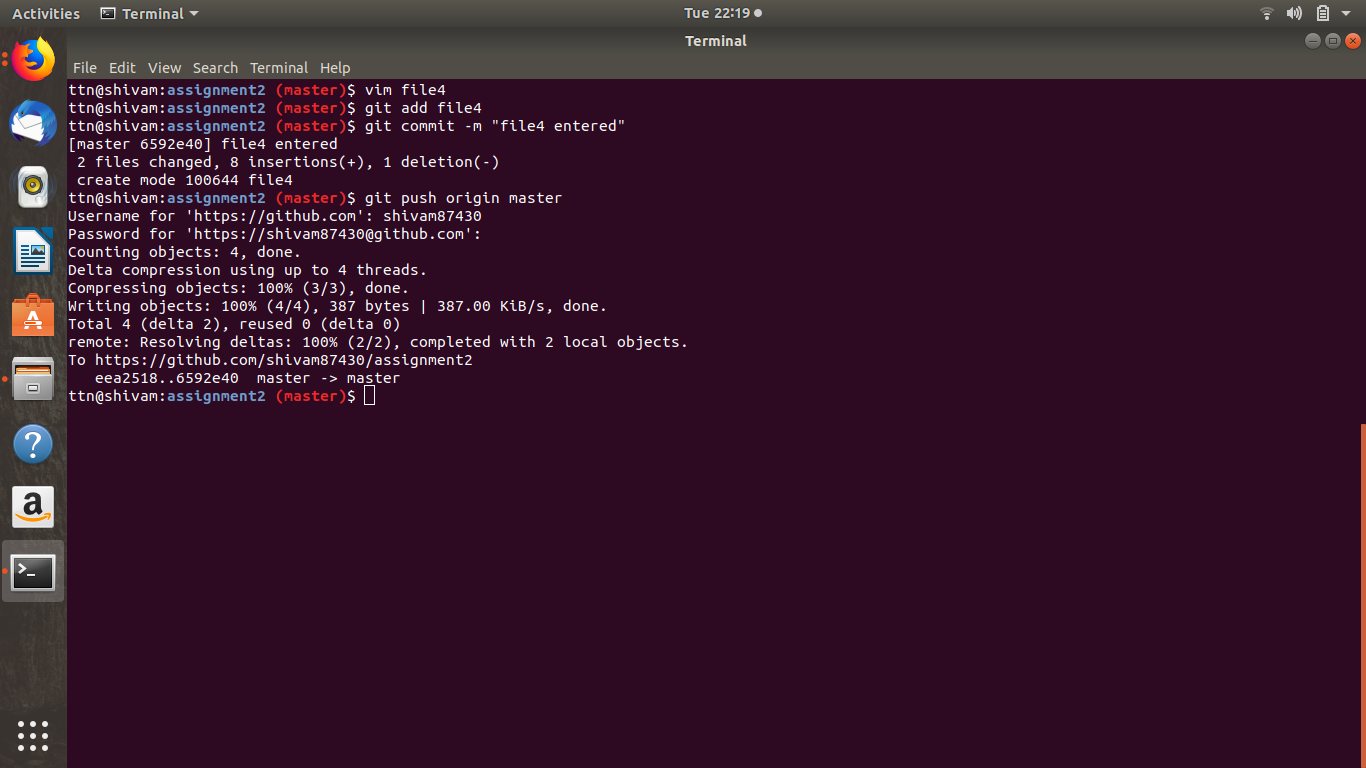
**Question 5 :** Commit the file

**Answer 5 :** $ vim file4

$ git add file4

$ git commit -m “file4”

$ git push origin master

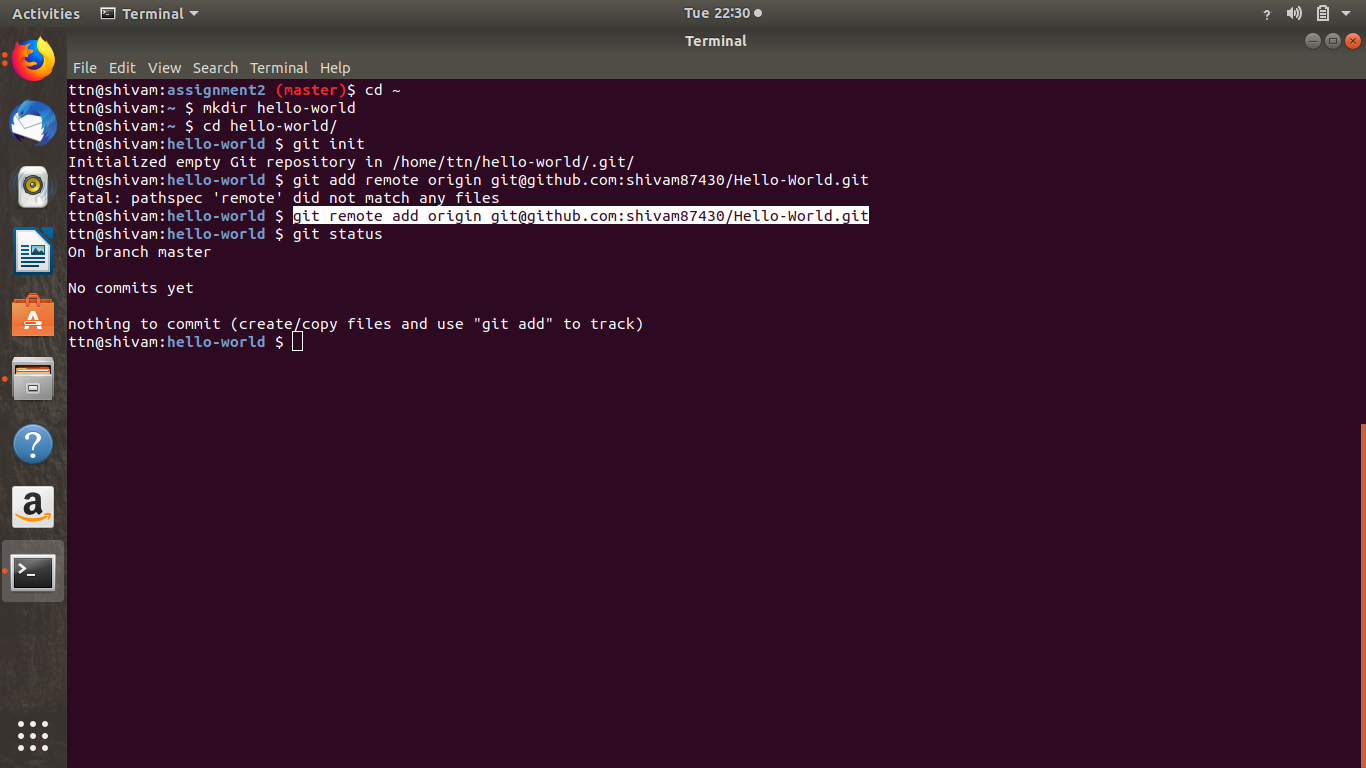


**Question 6 :** Add a remote

**Answer 6 :** command $ mkdir hello-world

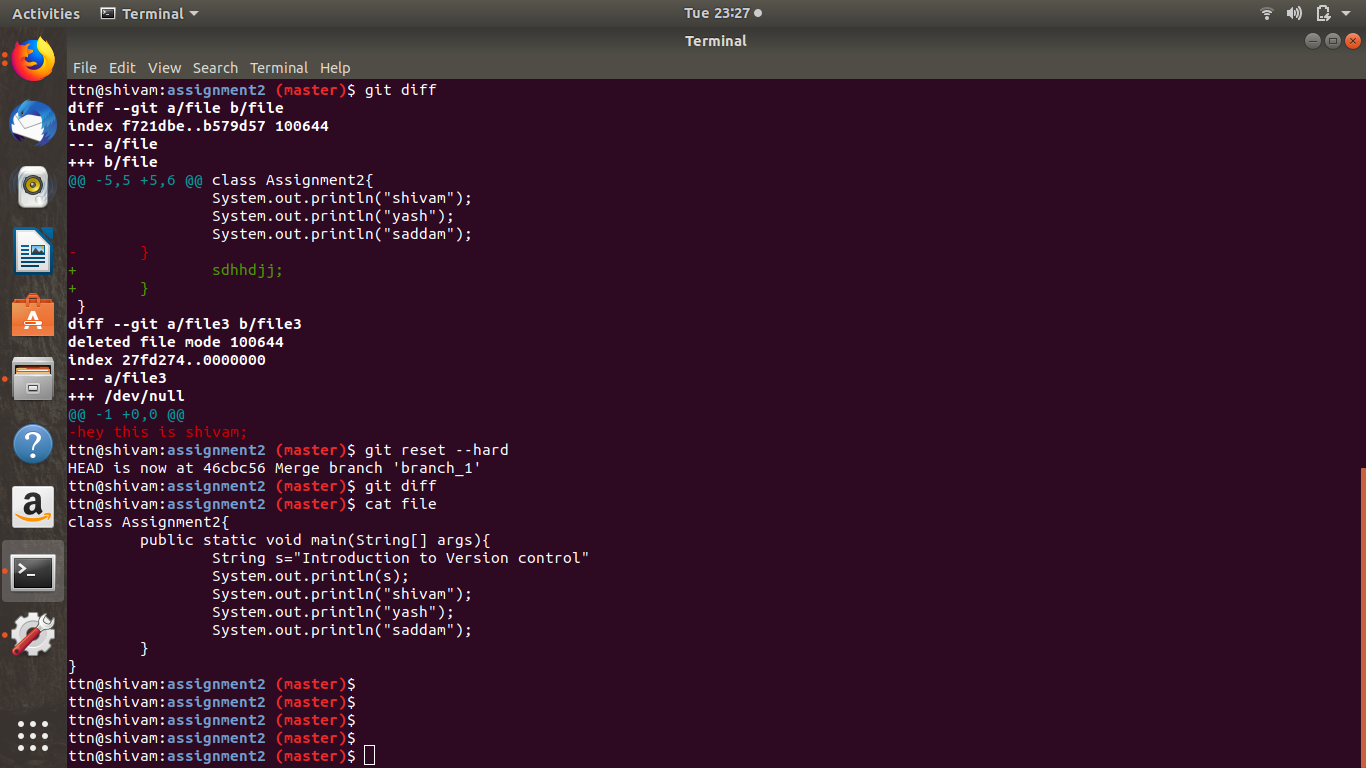
Command $ cd hello-world/

Command $ git remote add origin <https://github.com/shivam87430/hello-world.git>



**Question 7:** Undo changes to a particular file

**Answer 7 :**



**Question 8 :**Push changes to Github

**Answer 8 :** Push changes to Github

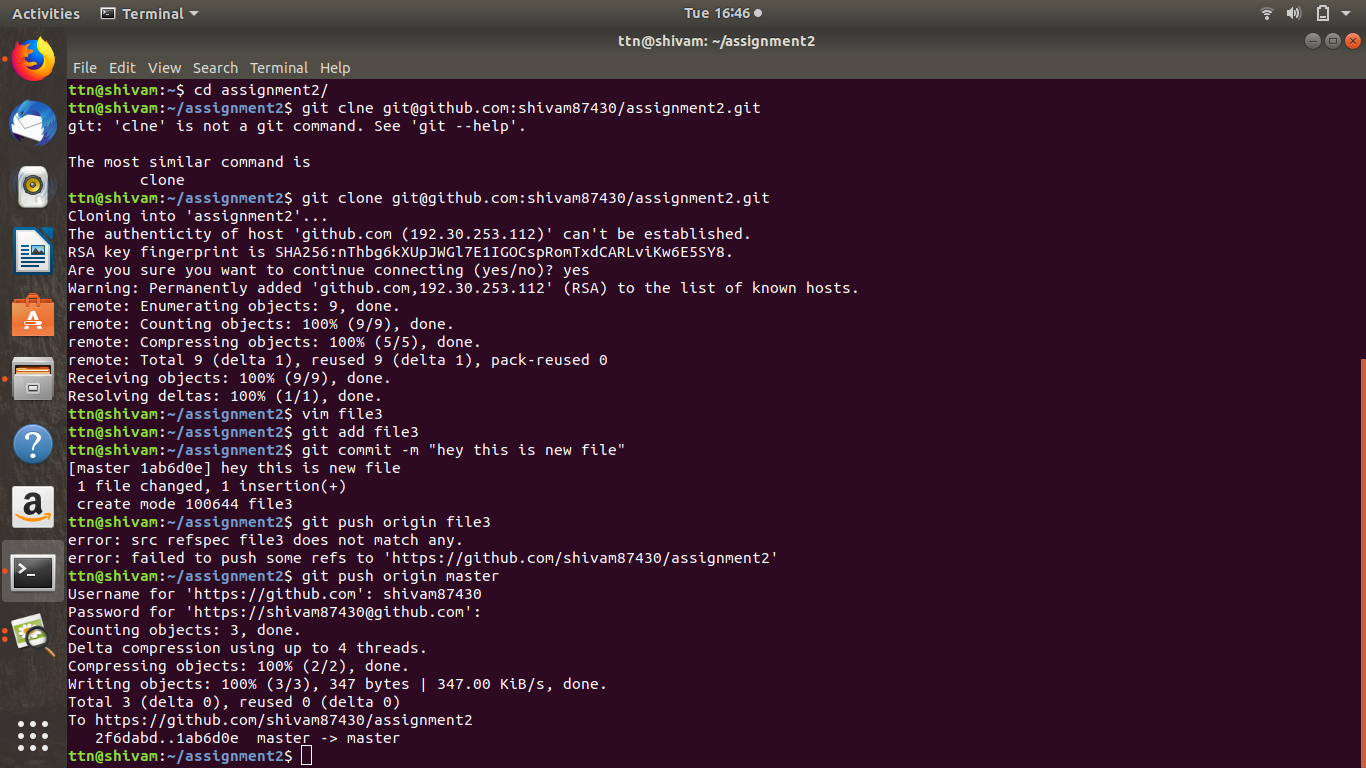
Vim file3

Write on file3

git add file3

git commit -m “message”

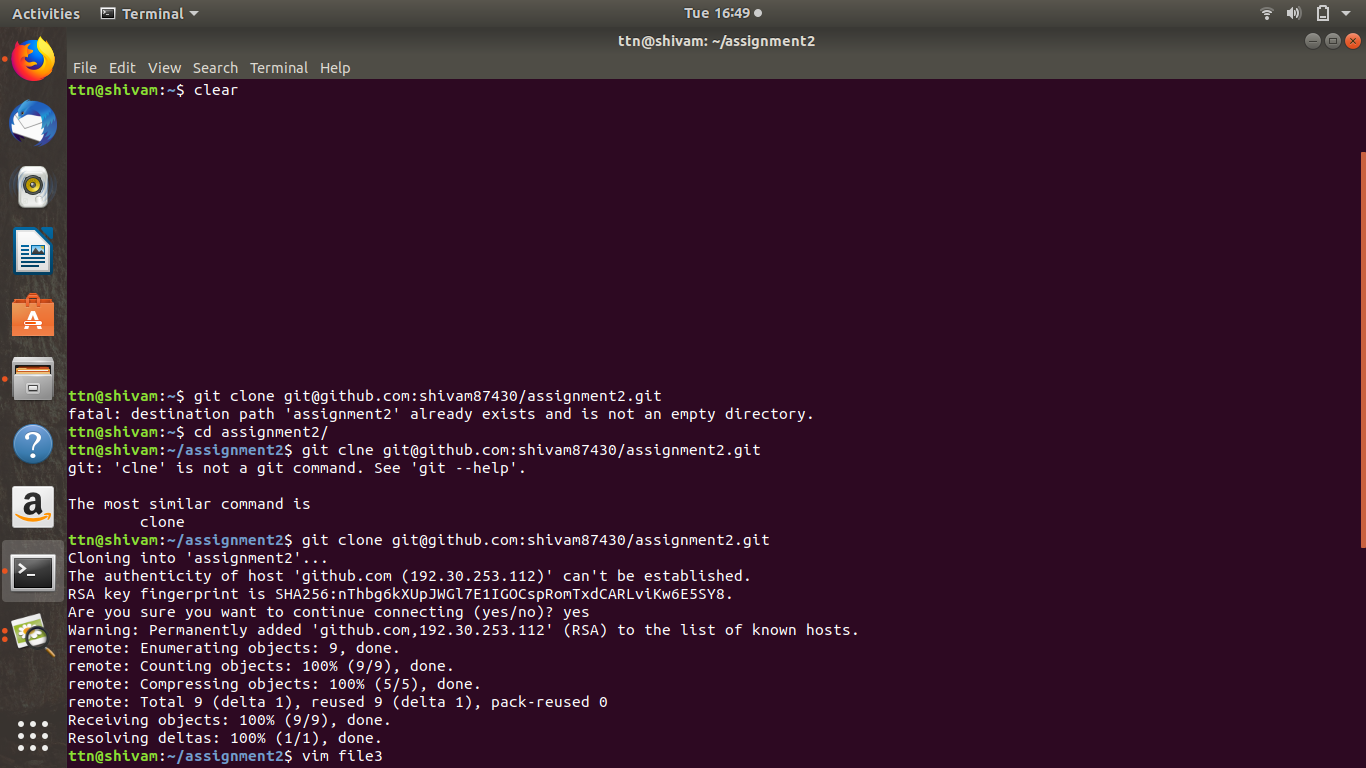
git push origin master



**Question 9 :**Clone the repository

**Answer 9 :** Clone the repository

git clone [git@github.com](mailto:git@github.com):shivam87430/assignment2.git



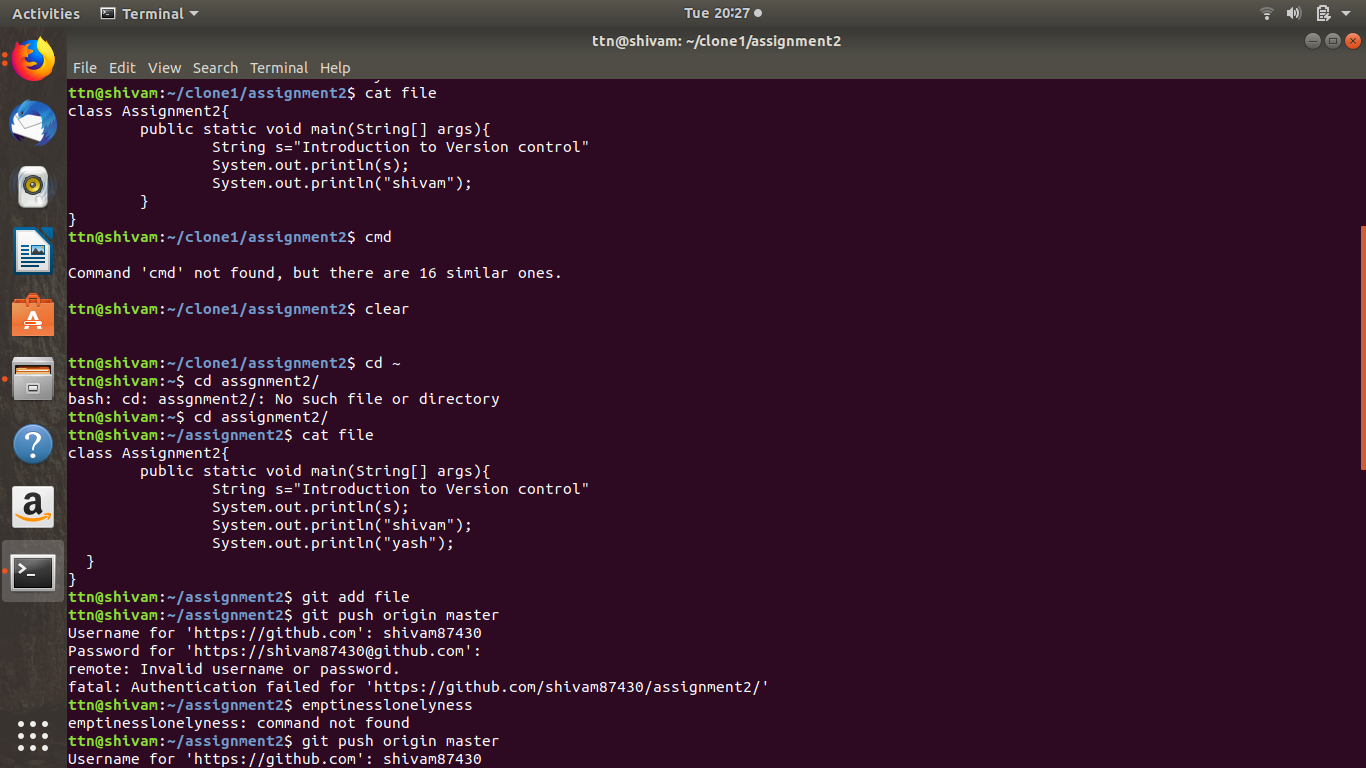
**Question 10 :** Add changes to one of the copies and pull the changes in the other.

**Answer 10 :** make new directory by command:- mkdir clone1

$ cd clone1/

$ git clone <url of repository>

$ cat file



After that do some change in ‘file’ file

Command vim file

: Wq that file

Add ‘file’ by command git add file

git commit -m “file changed”

git push origin master

After that change to remote directory

And pull that file by command:- $ git pull origin master





**Question 11:** Check differences between a file and its staged version

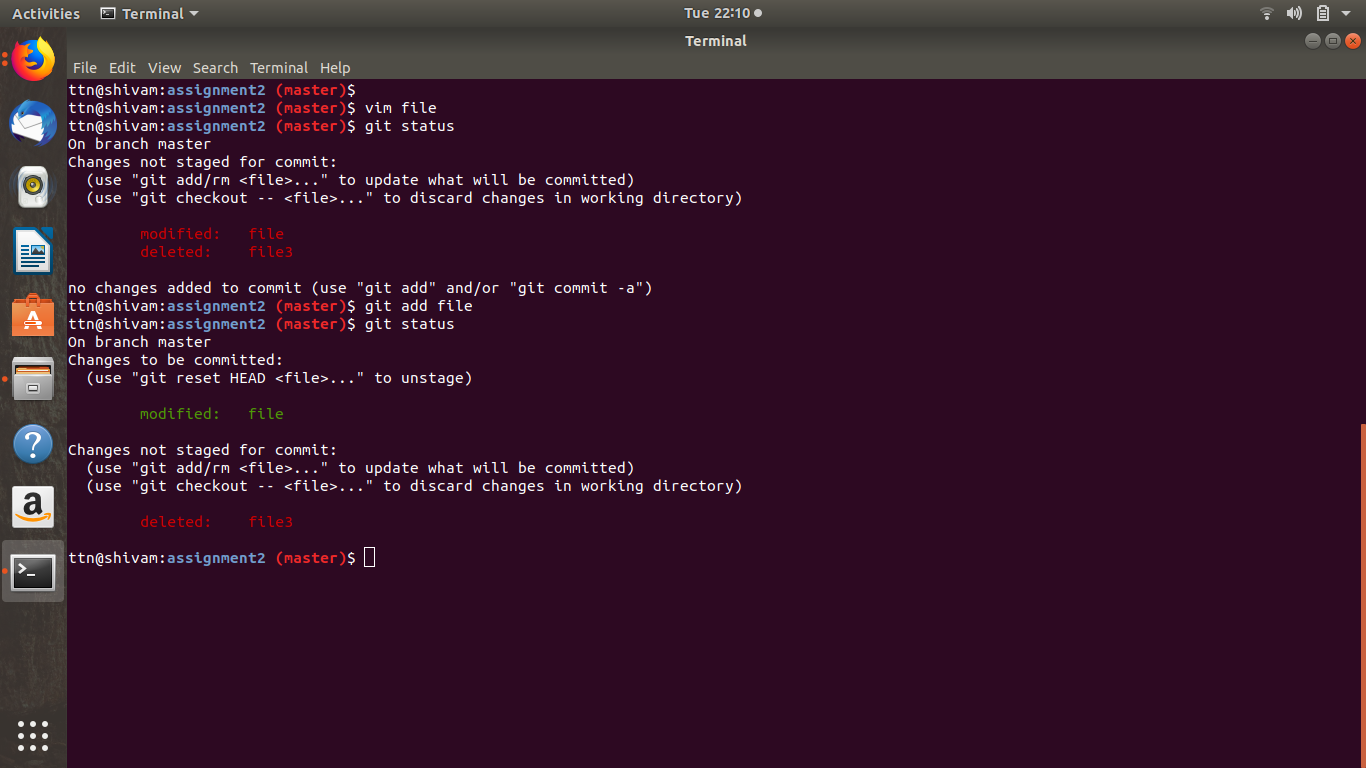
**Answer 11 :** Command $ vim file

Edit the file

Command $ git status

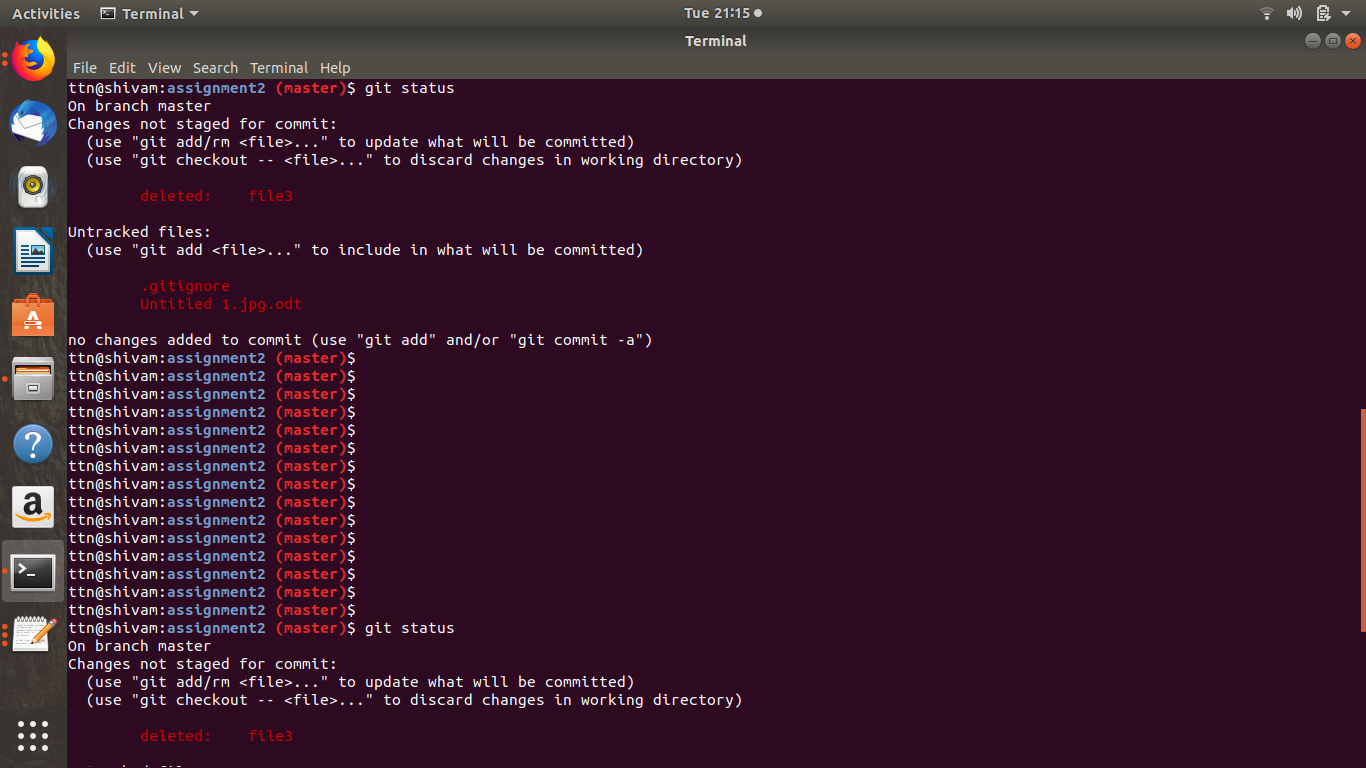
Now add the file Command : $ git add file

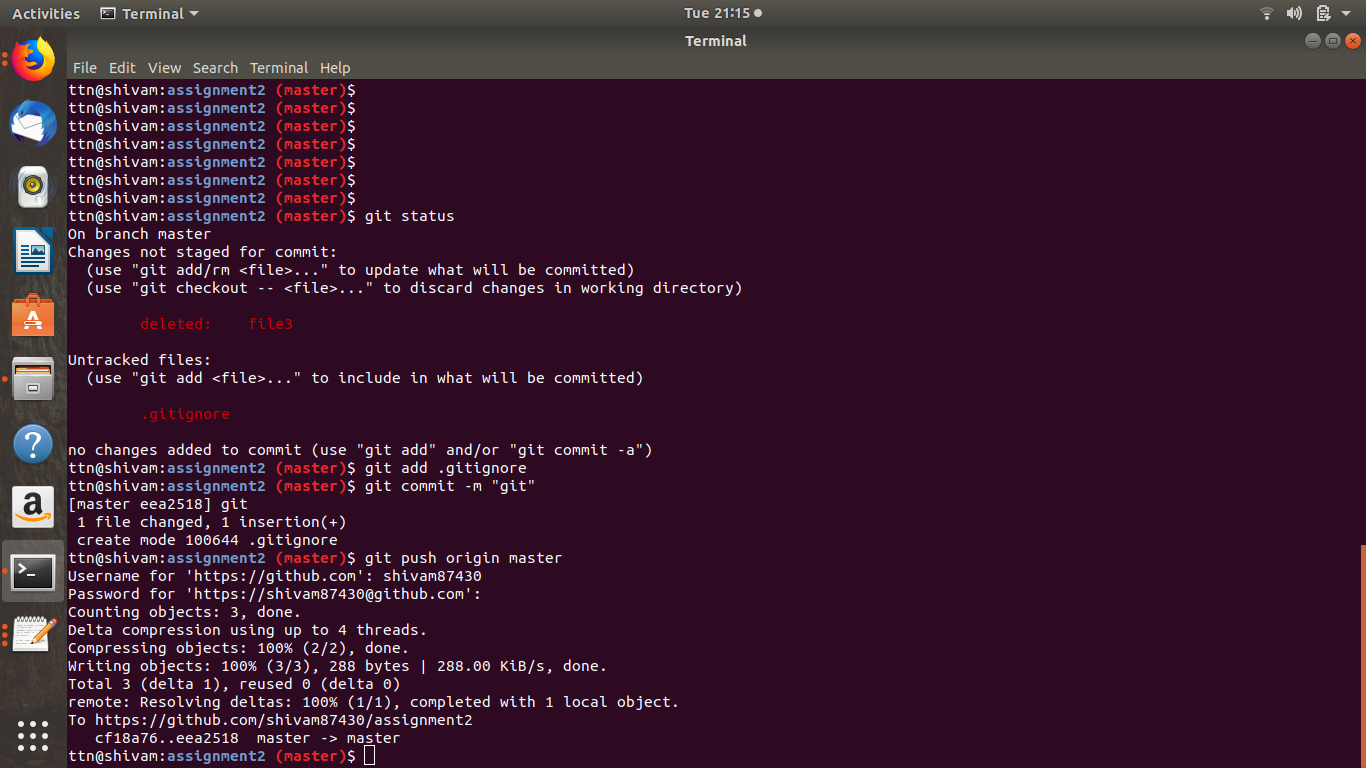
Command : $ git status



**Question 12:** ignore a few files to be checked in

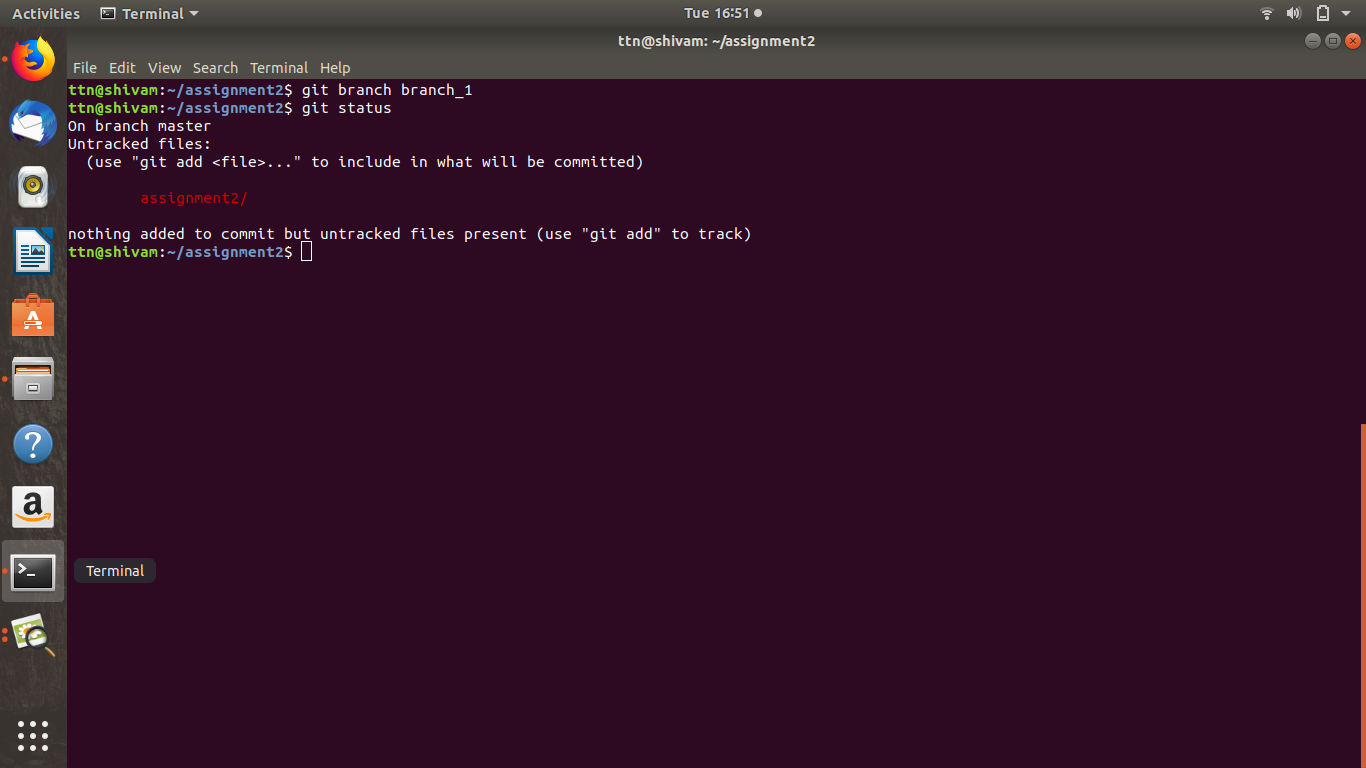
**Answer 12:** just put the files in .getignore file and save the .getignore file





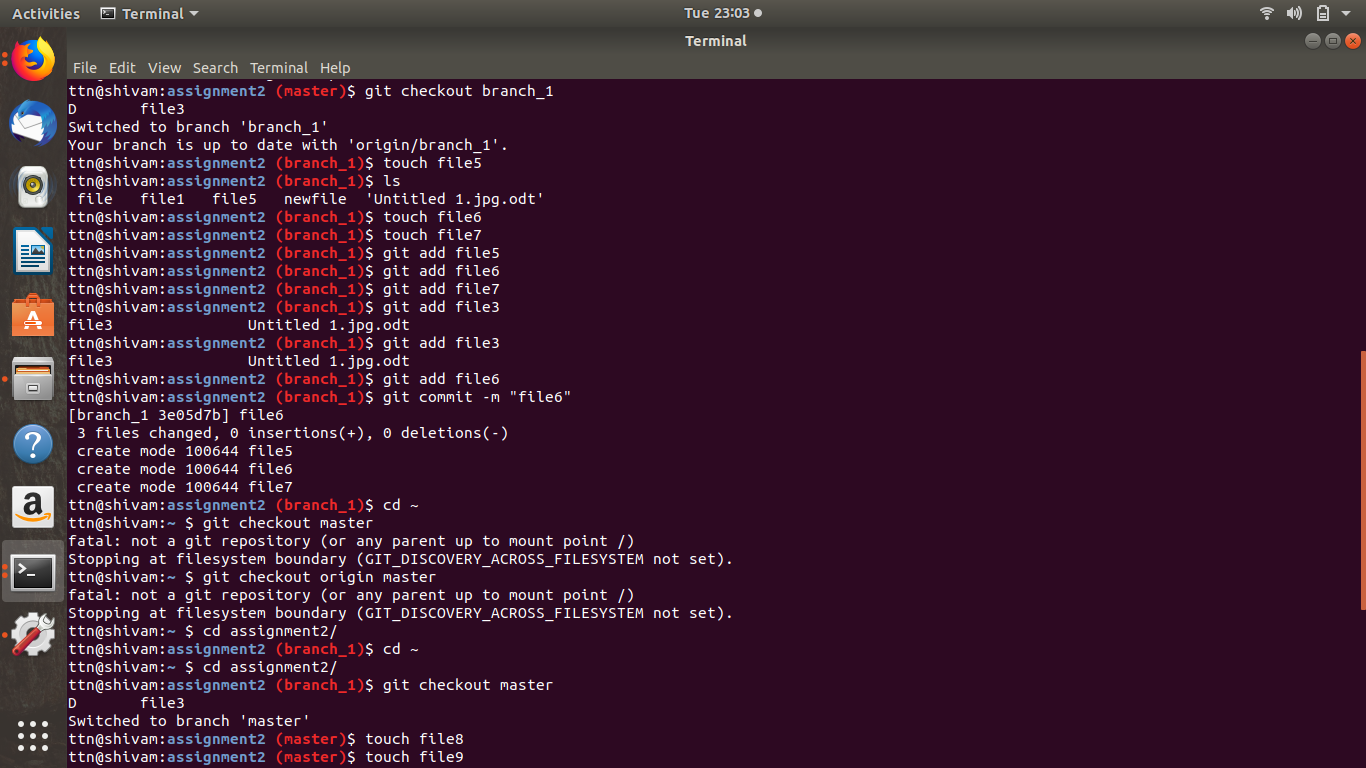
**Question 13 :** Create a new branch.

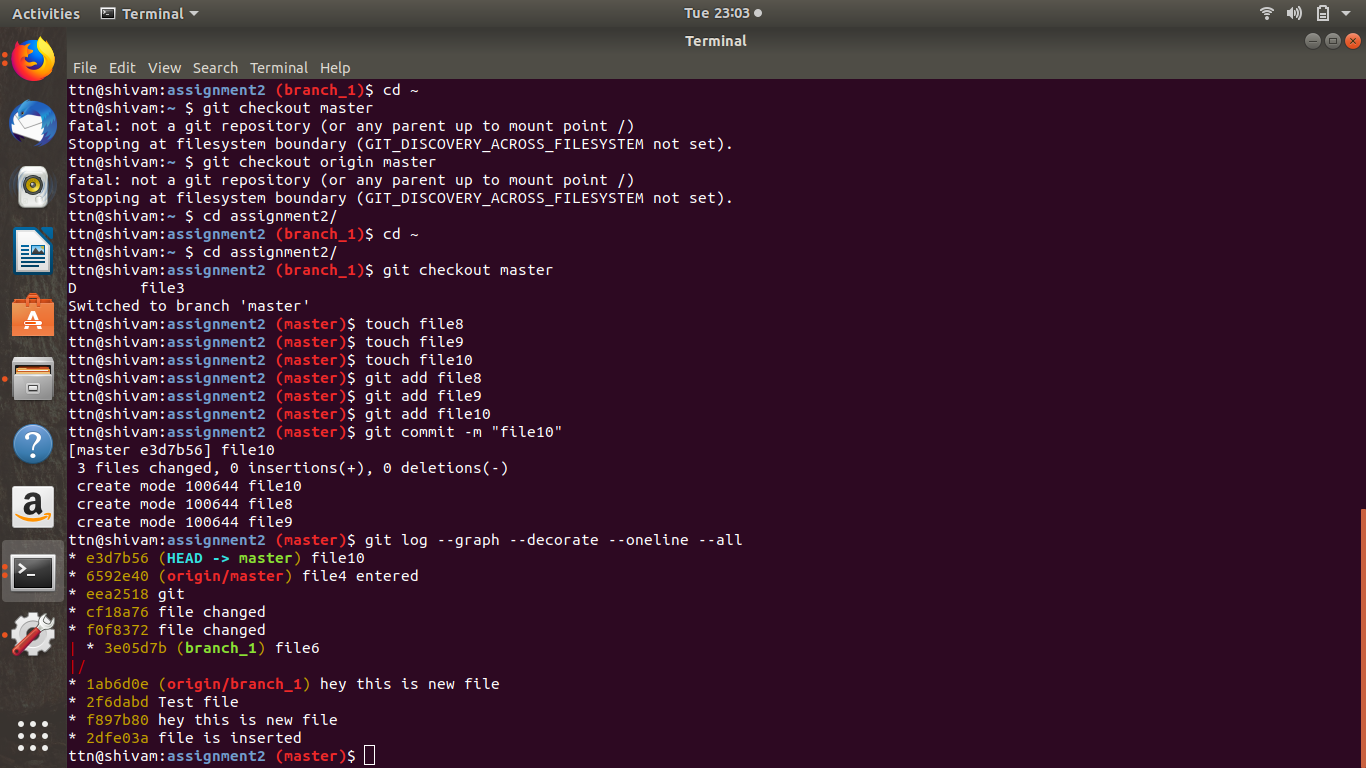
**Answer 13 :** command : git branch branch\_1



**Question 14 :** Diverge them with commits

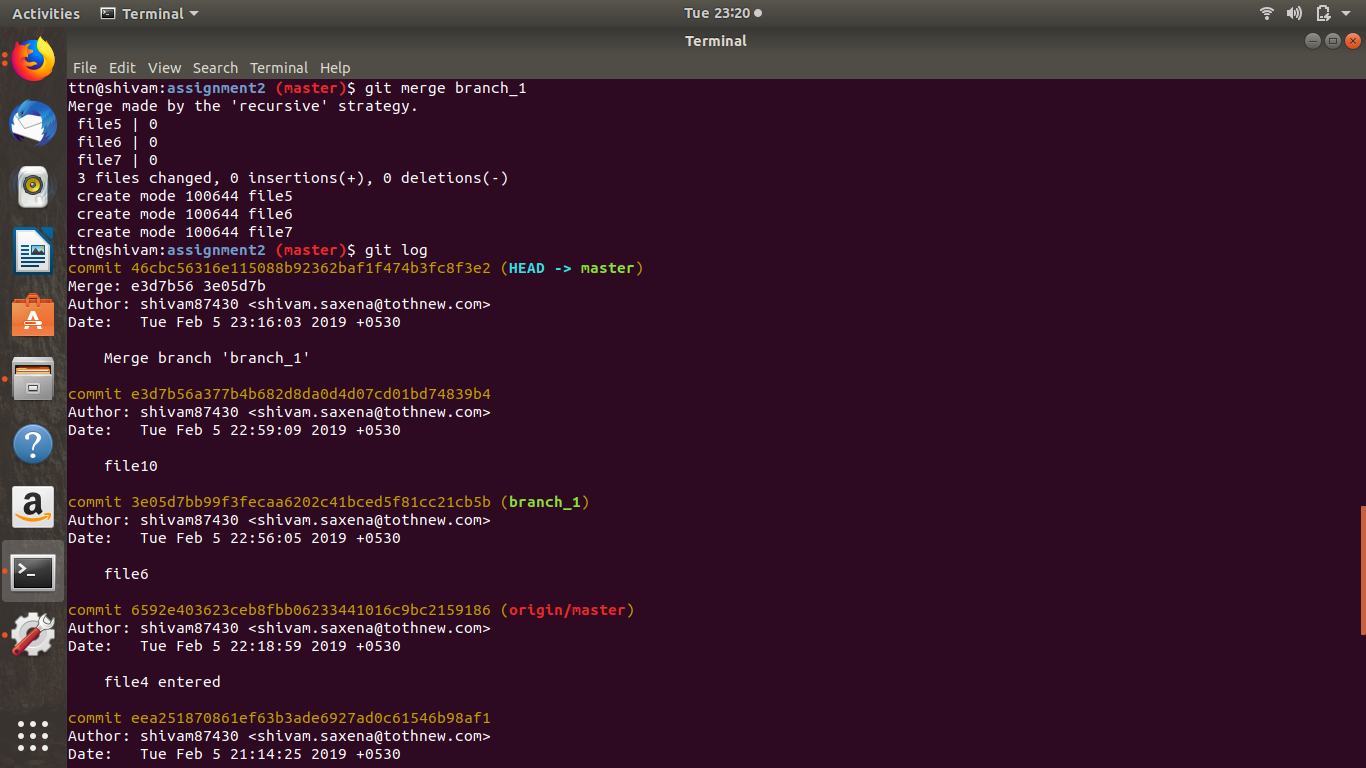
**Answer 14 :**

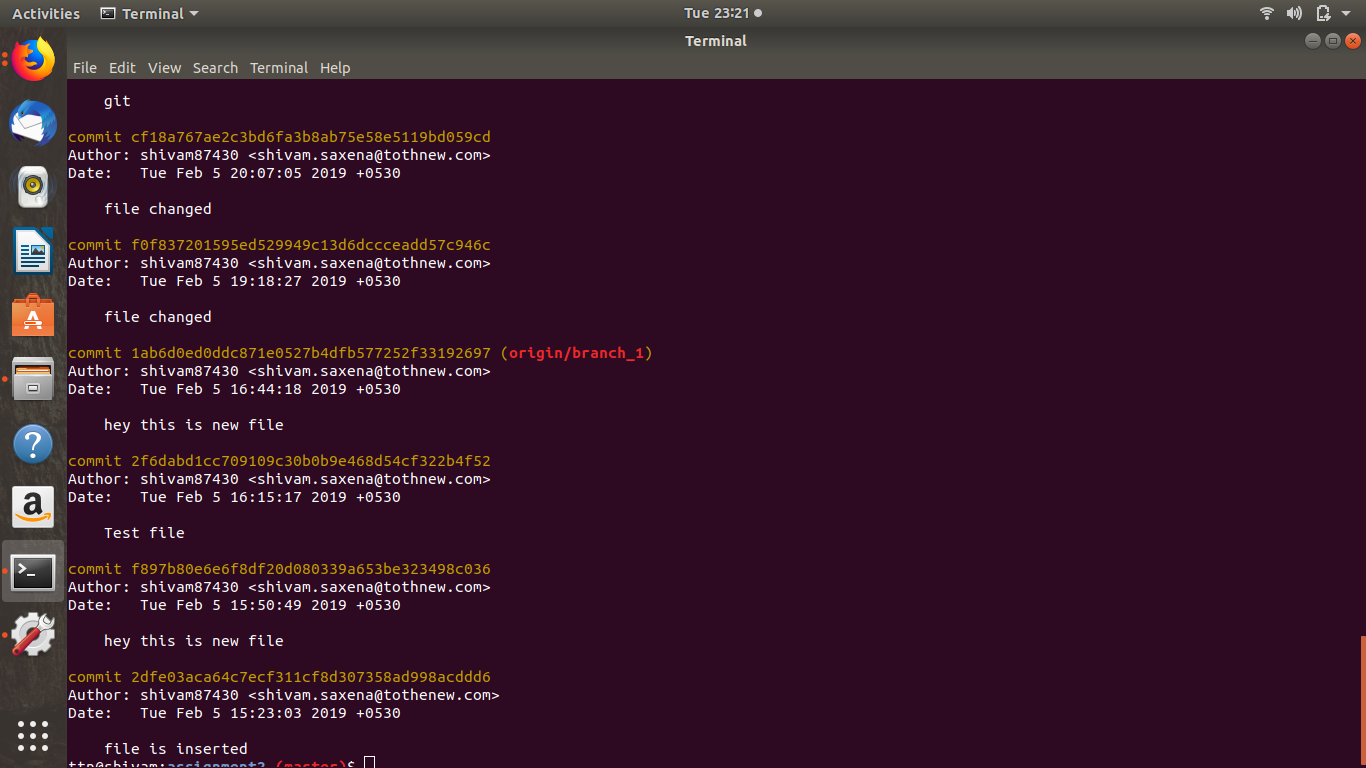




**Question 16 :**Try merging and resolve merge conflicts

**Answer 16 :**



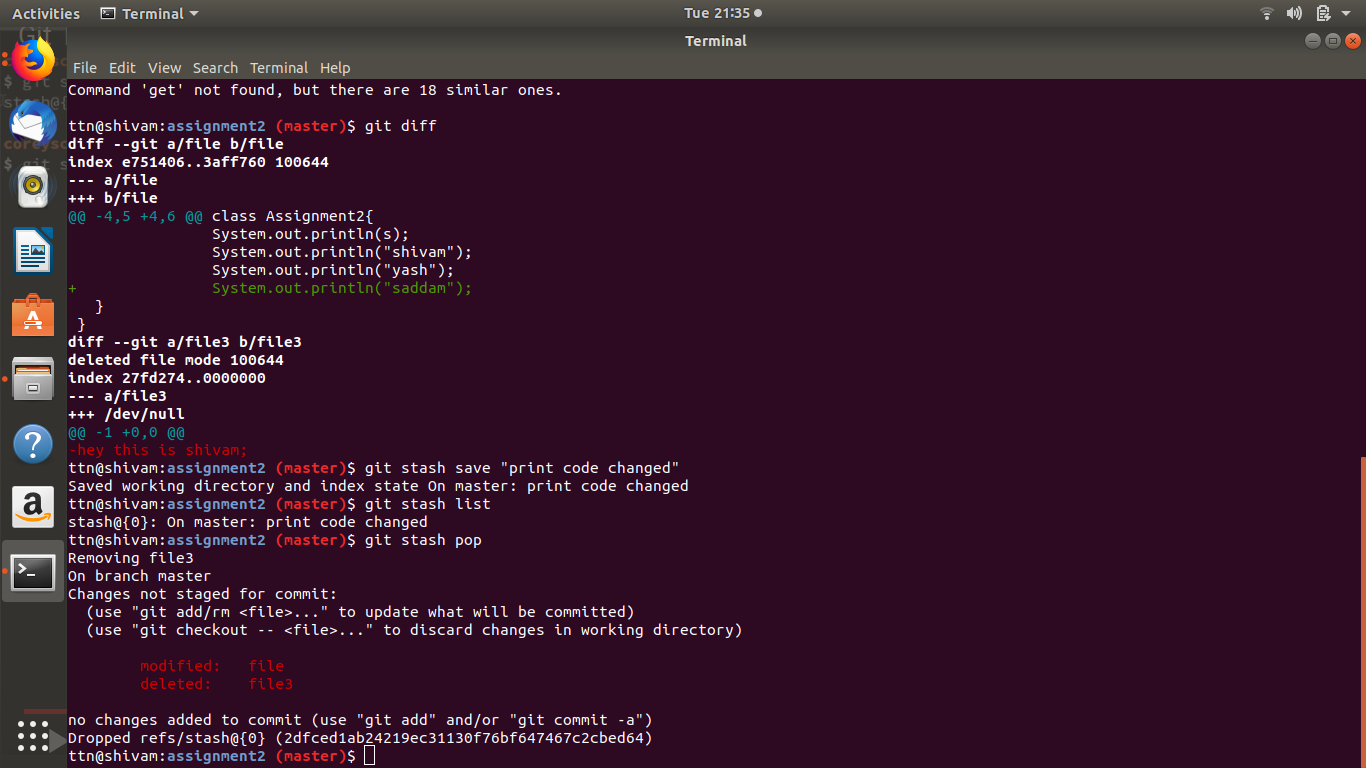
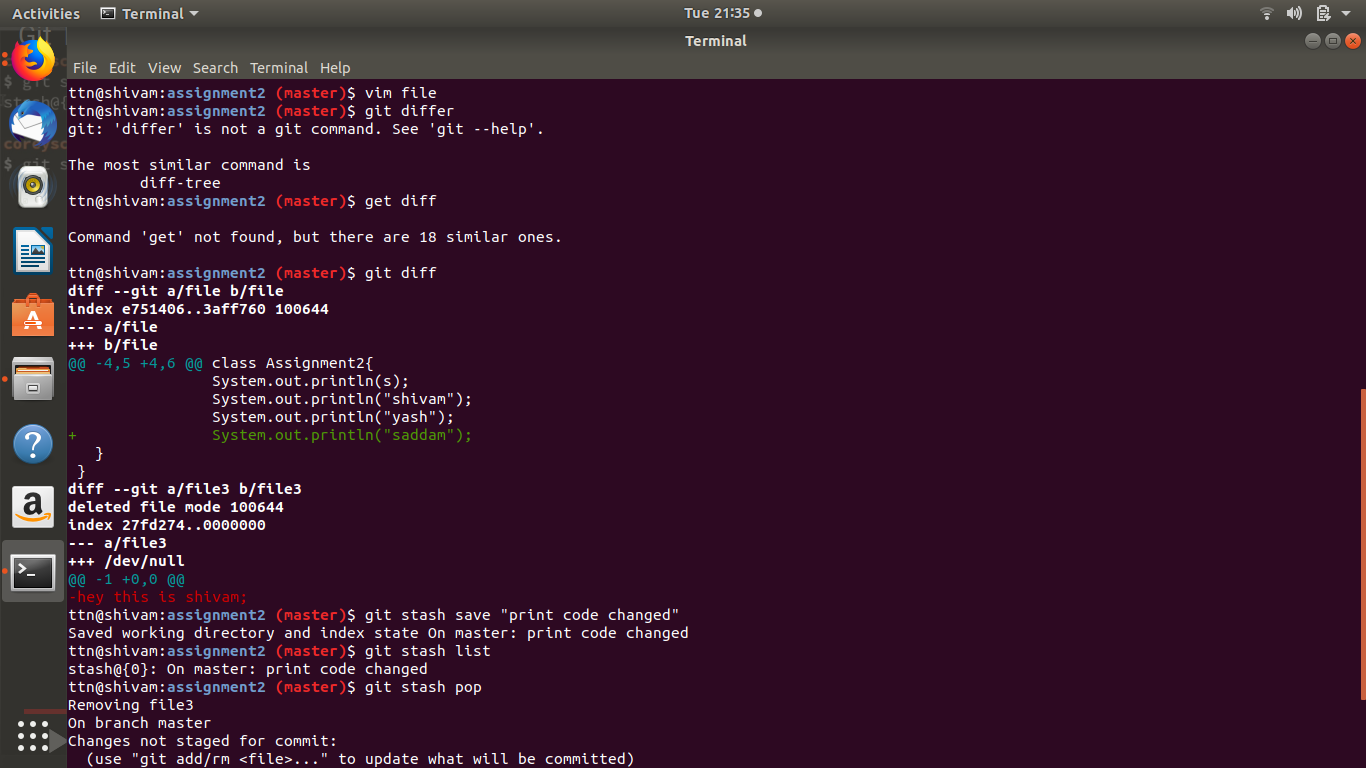


**Question 17 :** Stash the changes and pop them

**Answer 17 :**  git stash save “print code changed”

git stash list

git stash pop



**Question 18 :** 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 18 :**

press ctrl+h in master folder

Open .bashrc

Copy all above code and paste it at the last

Save the fle

Open new terminal