**Introduction to Version Control**

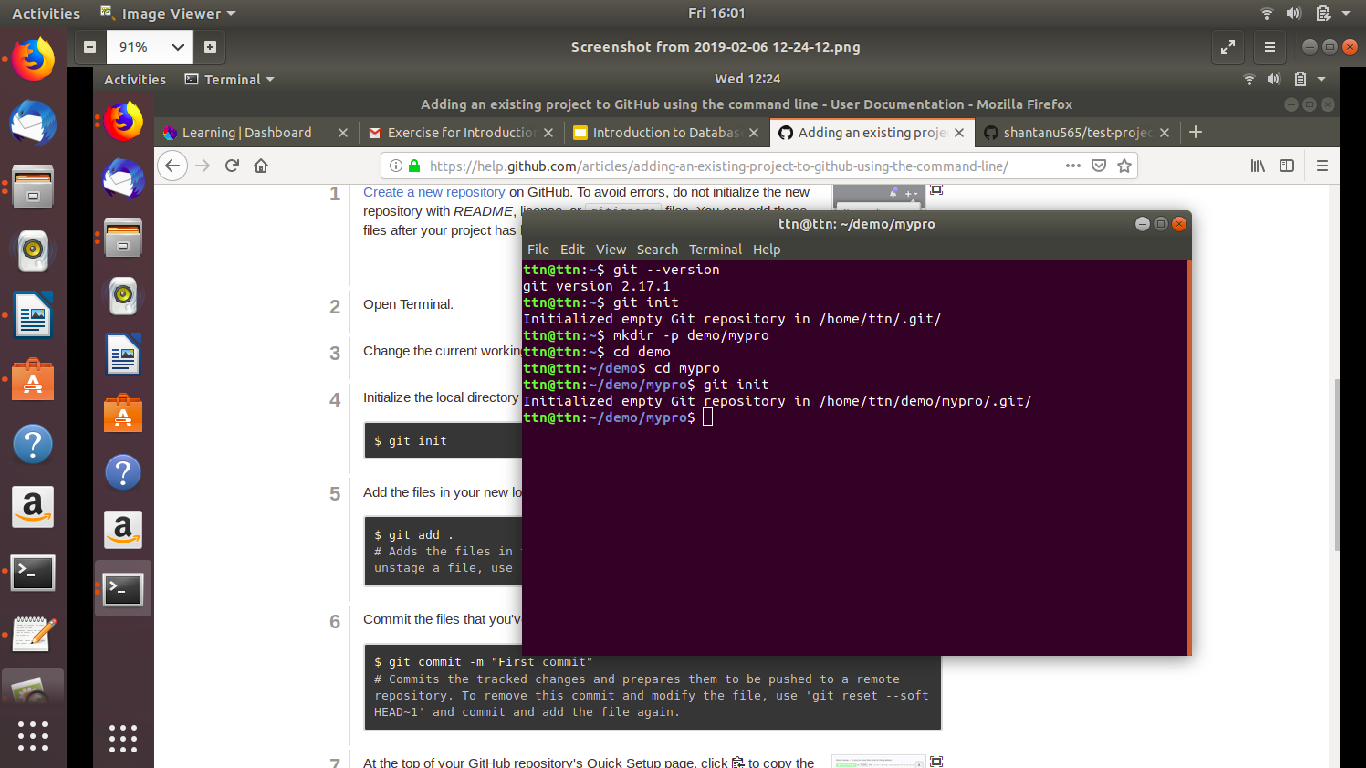
1. Git Setup<https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html>

Ans- 1)sudo apt-get install git //install git package

2)**git --version** //for git version

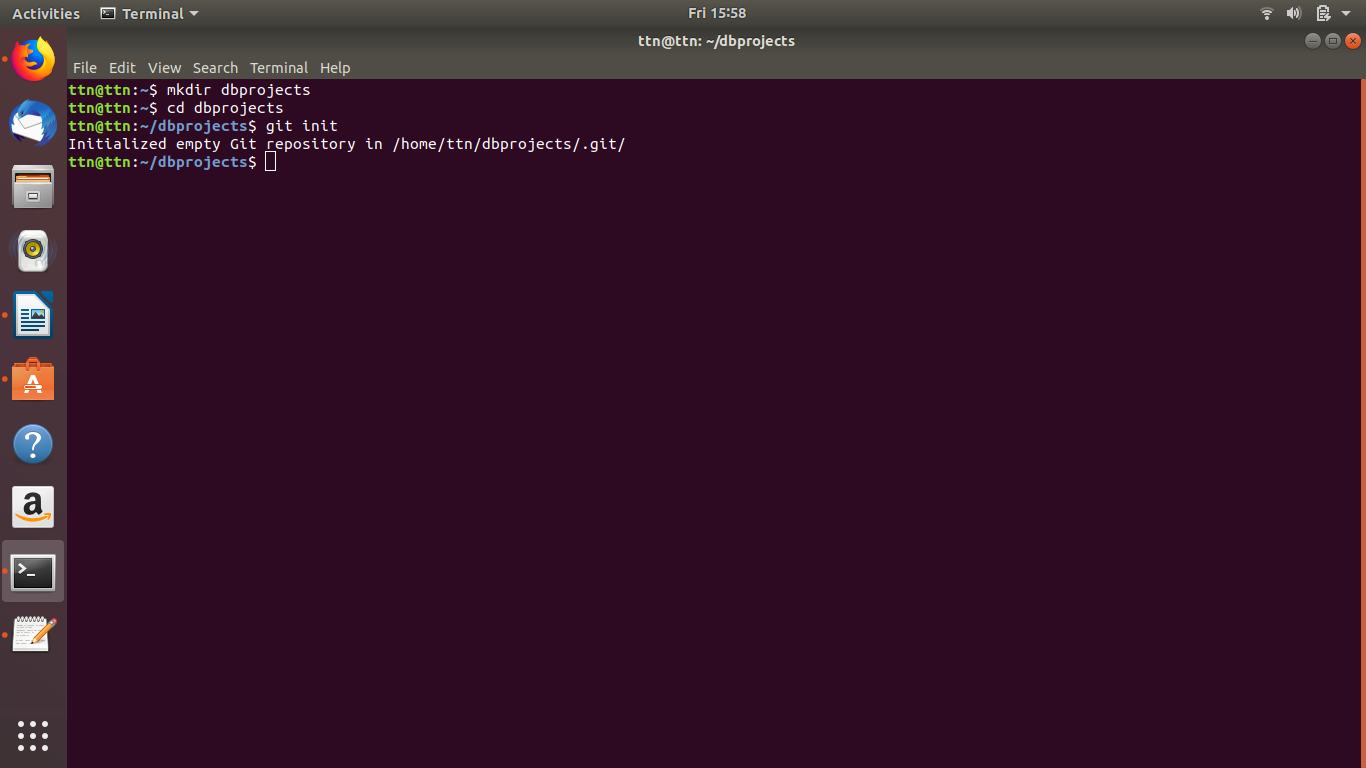
3)git config --global user.name<name>

git config --global user.email<email>

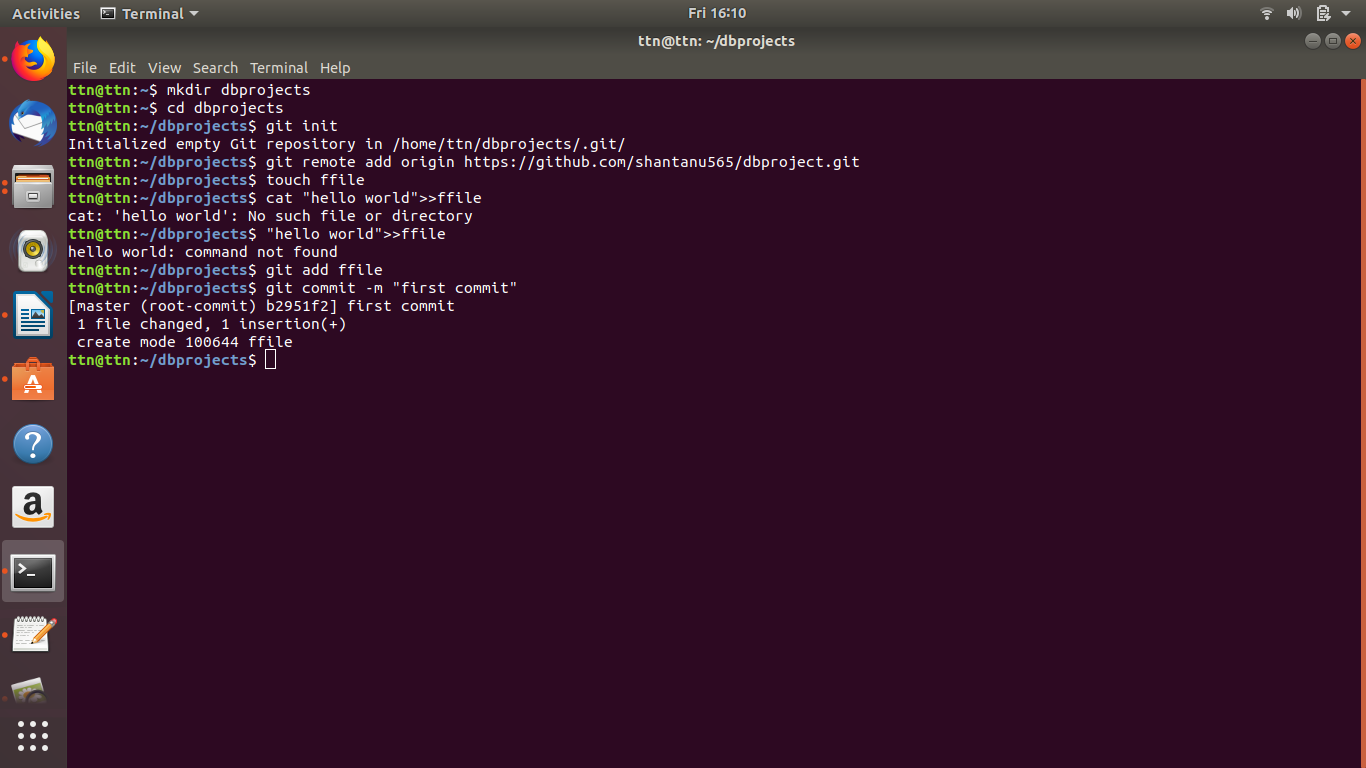


1. [Initialize a Git Repository](https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html)

Ans- **git init dbprojects**



1. Add files to the repository

Ans- **git add ffile**

1. Unstage 1 file

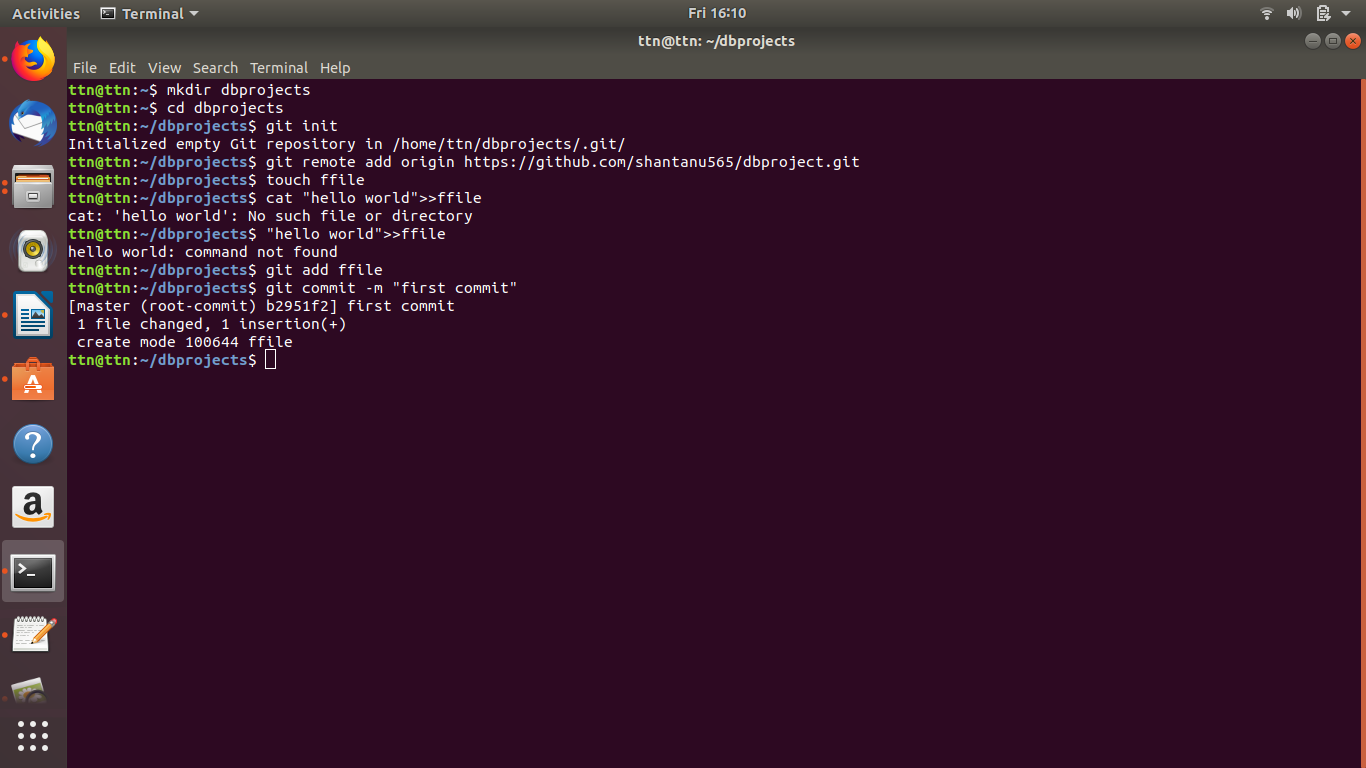
Ans- **To unstage a file we can use**

**1)git rm --cached file(this will makes git stop tracking the file completely,leaving it in file system)**

**OR 2)git reset HEAD --filename(this unstages any modifications made to the file since last commit but does not revert them in file system.**

1. Commit the file

Ans- **git commit -m “first commit”**



1. Add a remote

Ans- **git remote add origin** [**https://github.com/shantanu565/dbproject.git**](https://github.com/shantanu565/dbproject.git)

**Screenshot(Same as Q5)**

**//to check if remote repository is added or not**

**Git remote -v**

1. Undo changes to a particular file

Ans- **git reset HEAD filename** -this will unstage the file to current commit

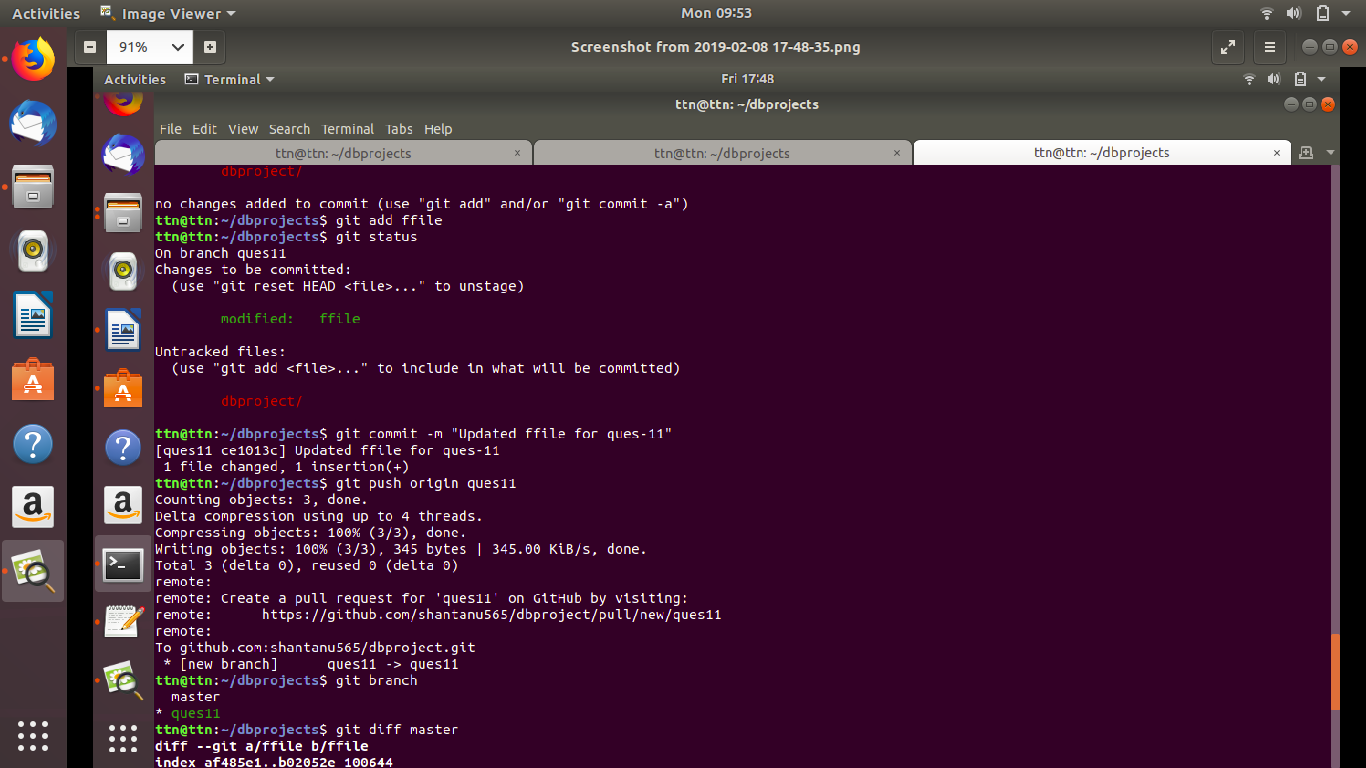
**git reset**  -this will unstage all the files but changes will be retained

**git stash**  -this will discard all local changes,but they will be saved for later.

**git reset hard**  -to discard every change permanently

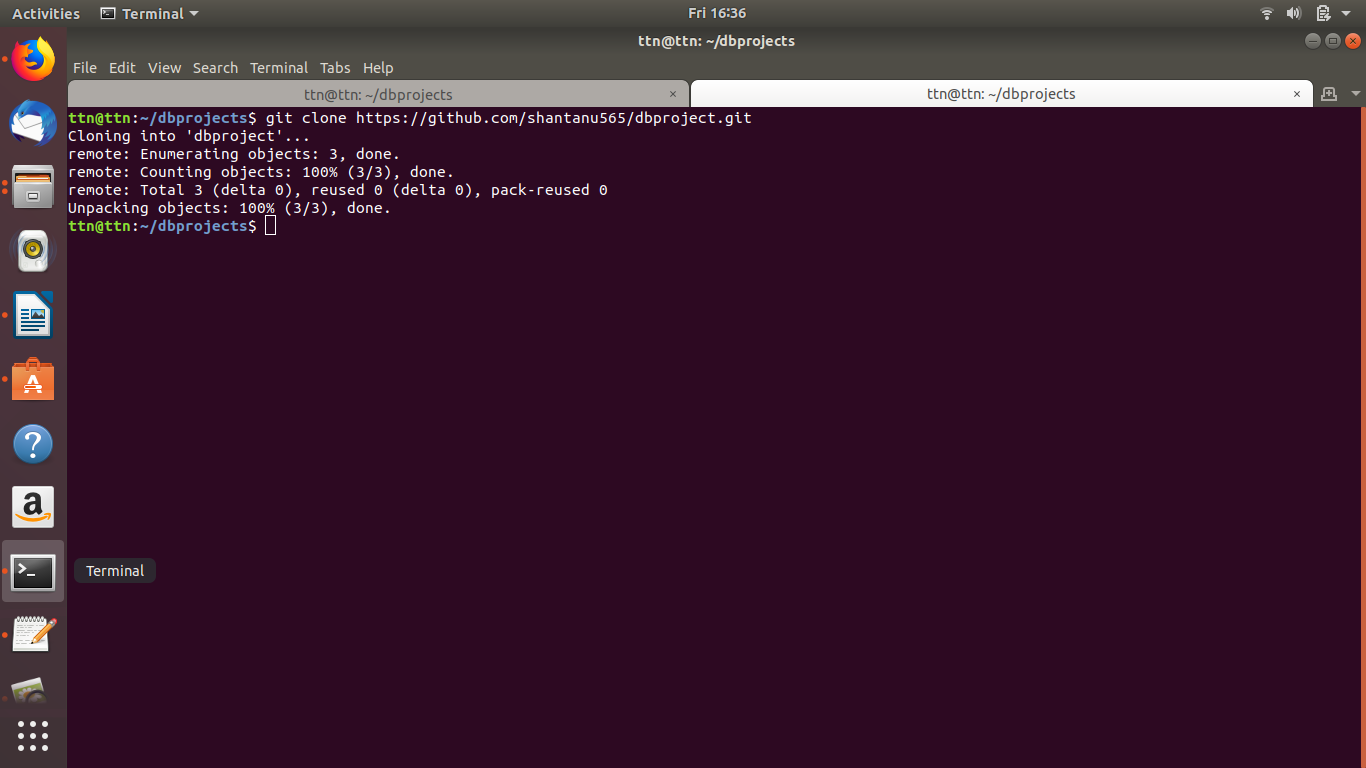
1. Push changes to Github

Ans- **git push**

****

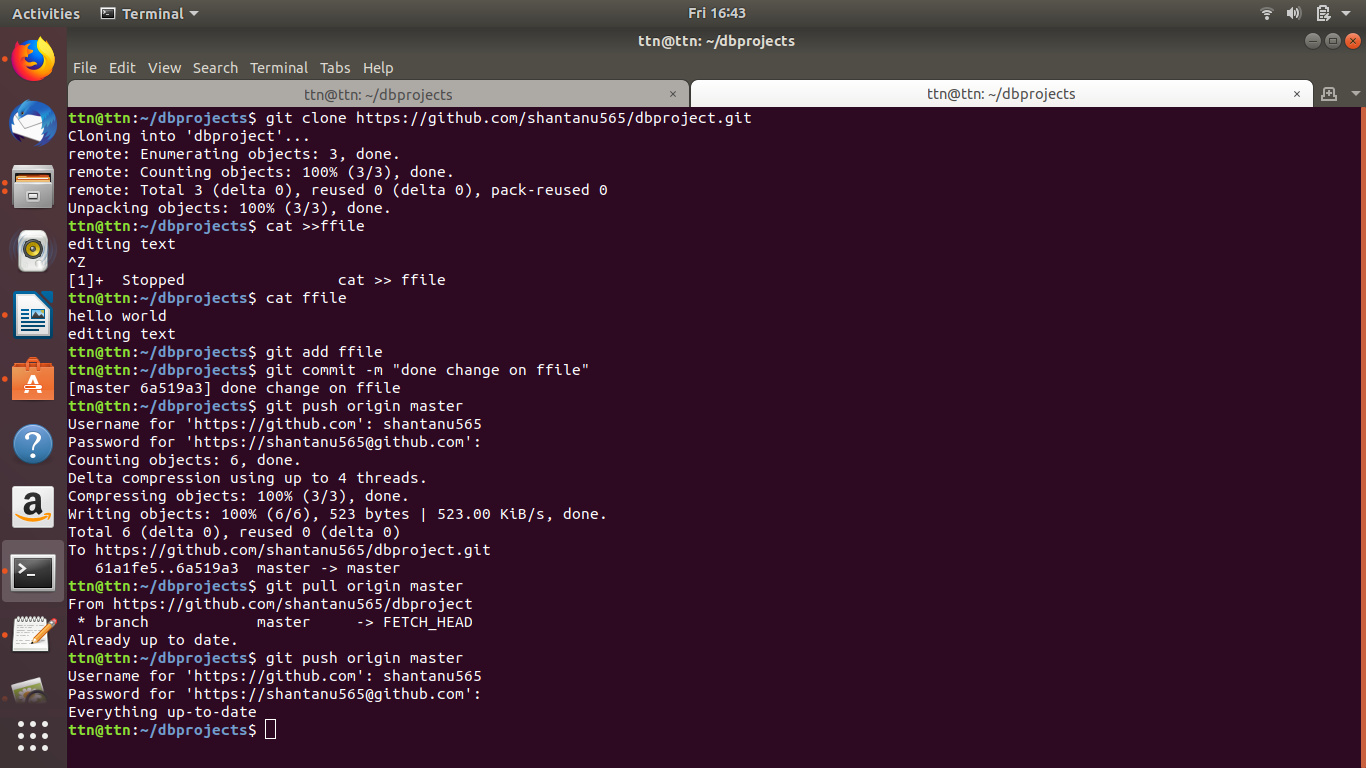
1. Clone the repository

Ans- **git clone** [**https://github.com/shantanu565/dbproject.git**](https://github.com/shantanu565/dbproject.git)

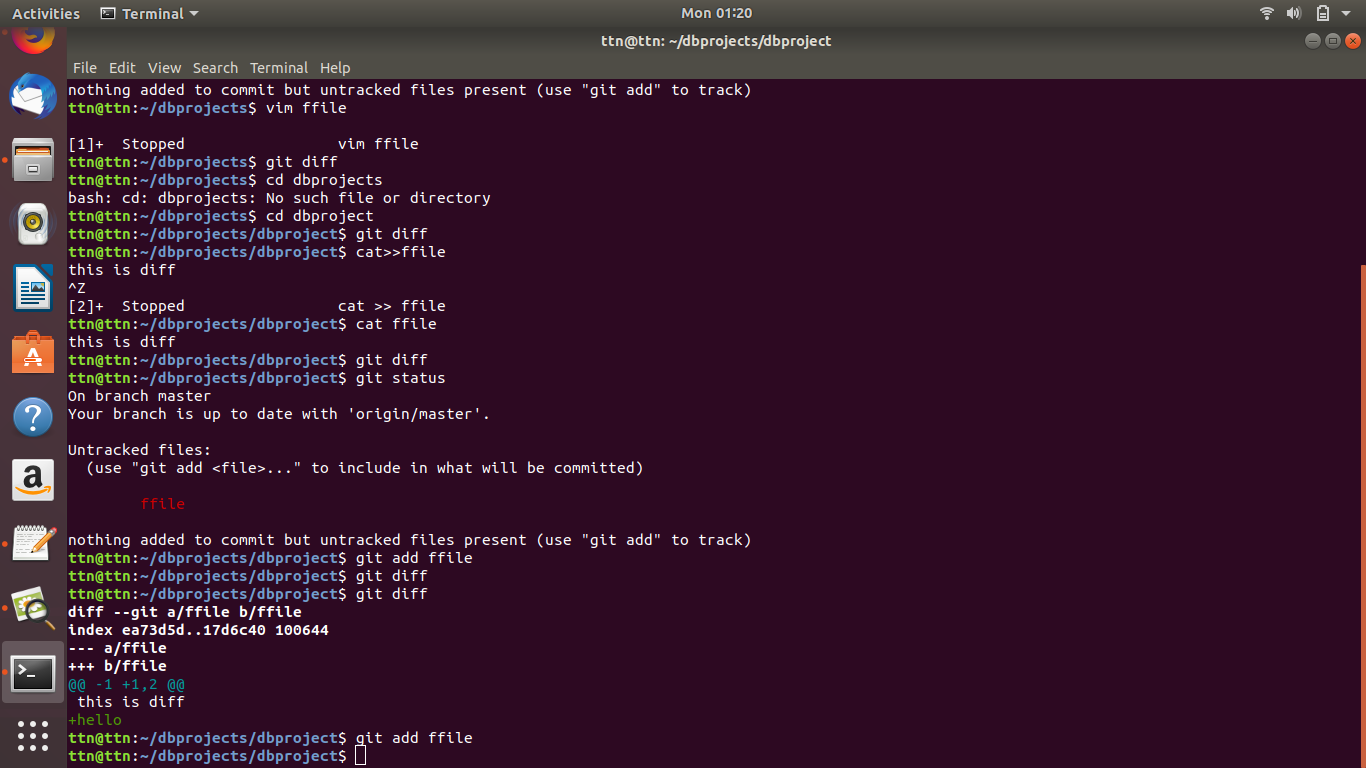


1. Add changes to one of the copies and pull the changes in the other.

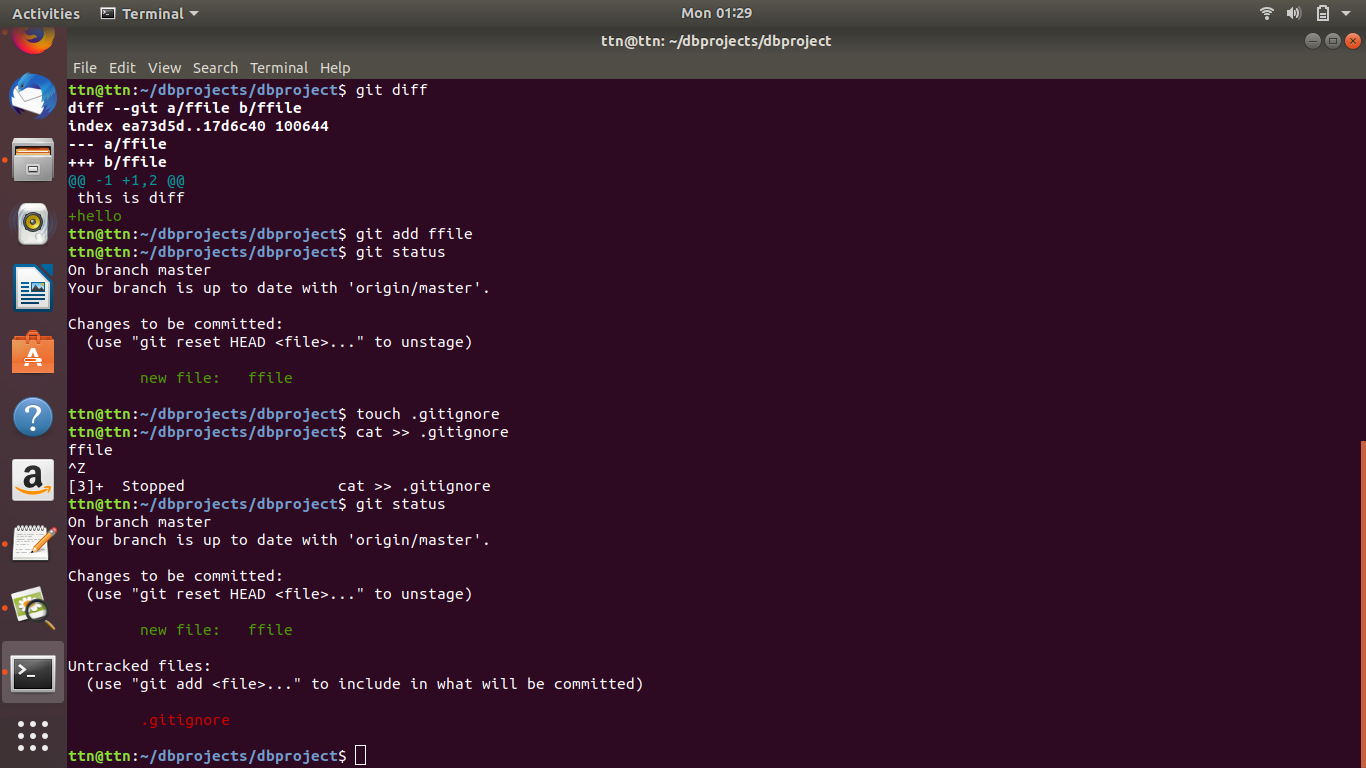
Ans-



1. Check differences between a file and its staged version

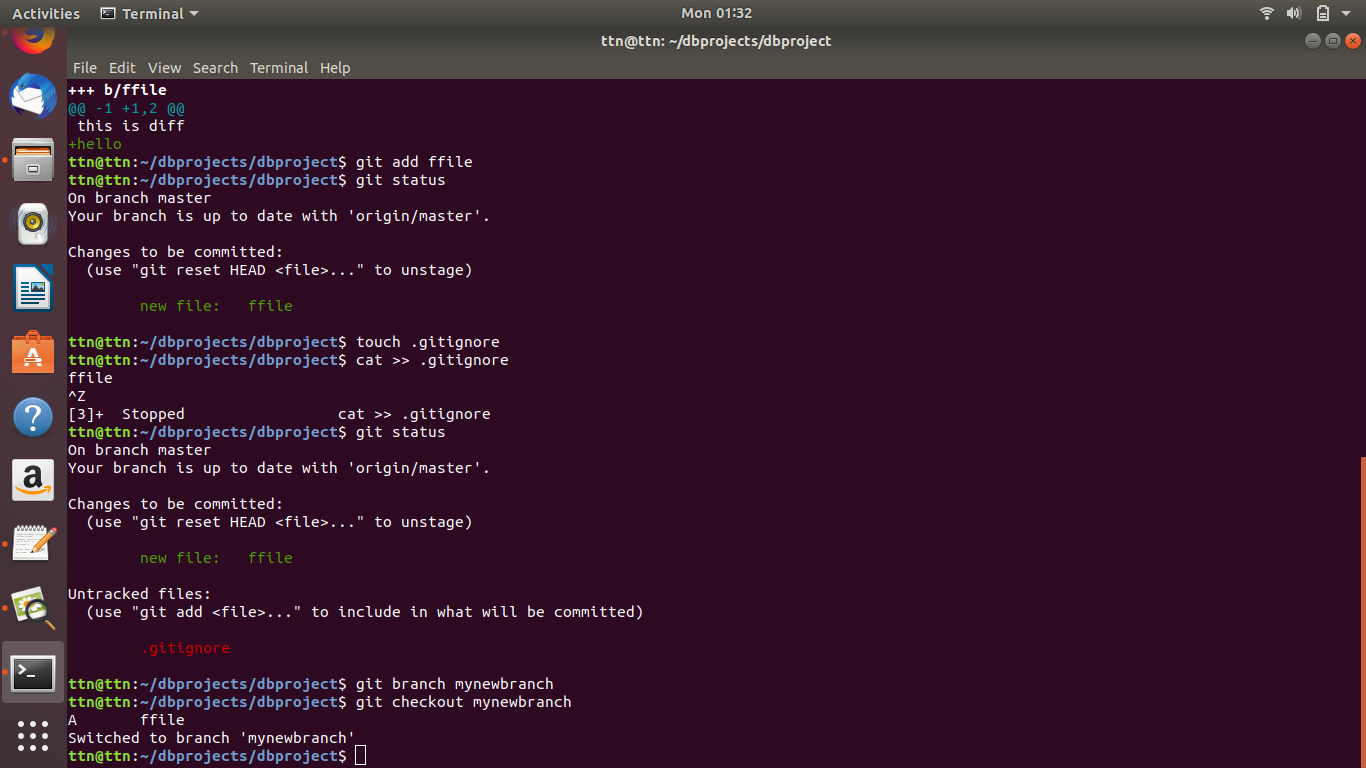
Ans- **git diff**

1. Ignore a few files to be checked in

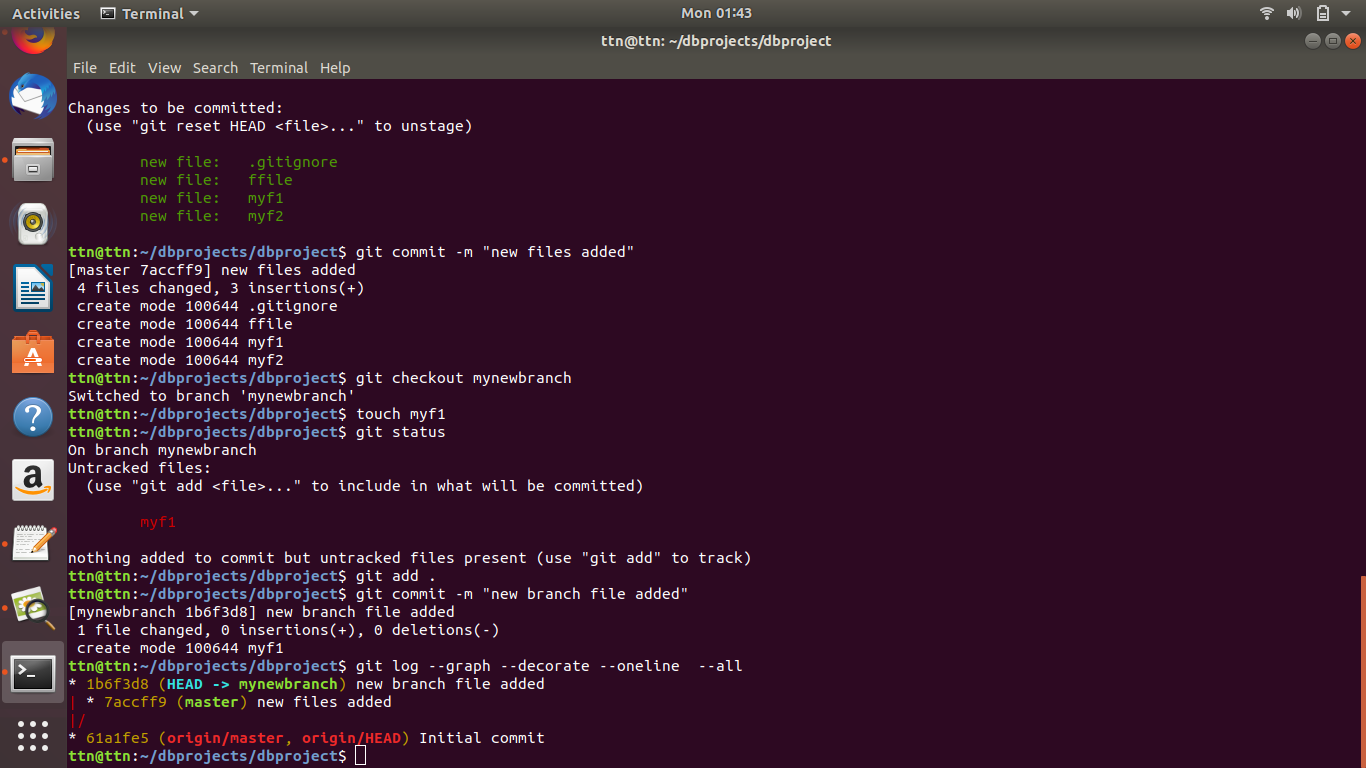
Ans- 

1. Create a new branch.

Ans- **git branch mynewbranch, git checkout mynewbranch**



1. Diverge them with commits

Ans- 

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

Ans- **Same as Q14**

16.Try merging and resolve merge conflicts

Ans-1)cd dbproject

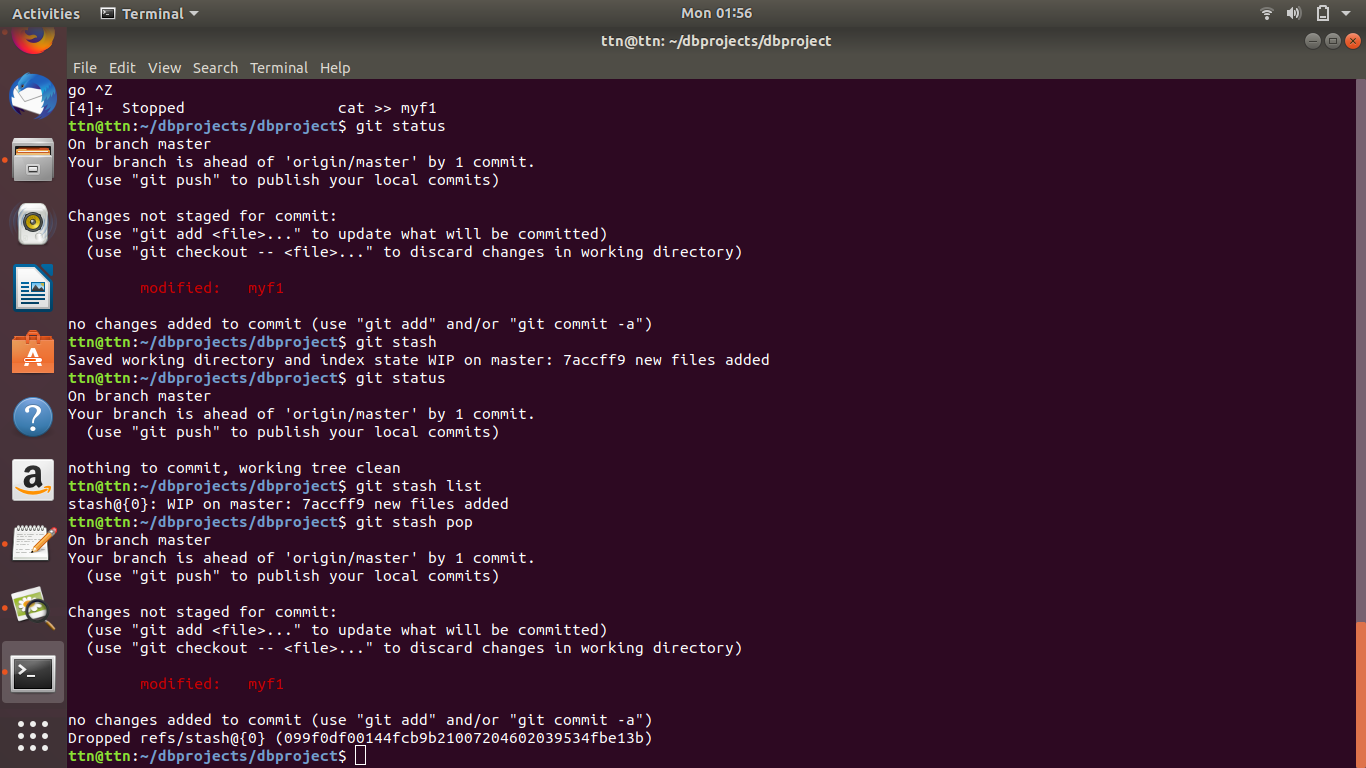
2)git status

3)Navigate to file & decide if you want to keep only your branch's changes, keep only the other branches changes, or make a brand new change, which may incorporate changes from both branches. Delete the conflict markers <<<<<<<, =======, >>>>>>> and make the changes you want in the final merge.

4)git add .

5)git commit -m “resolved merge conflict”

17.Stash the changes and pop them

Ans- 

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

Ans- 