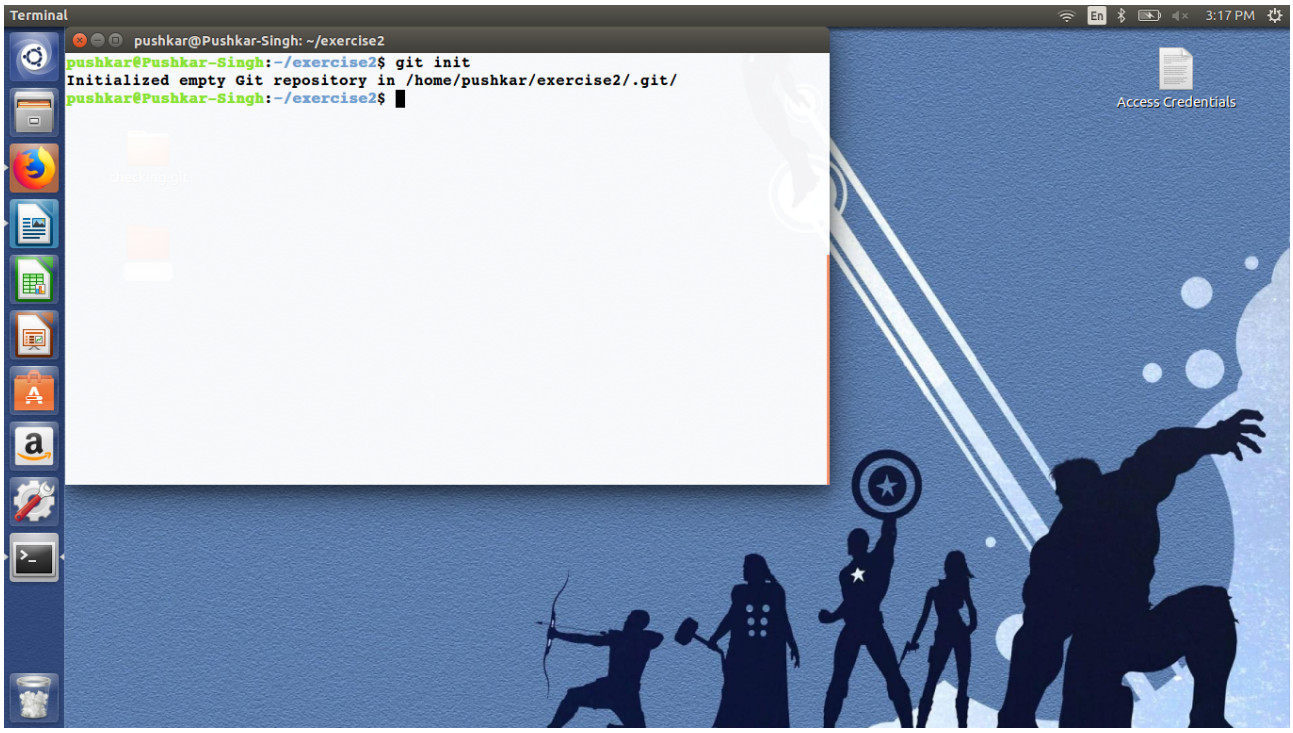
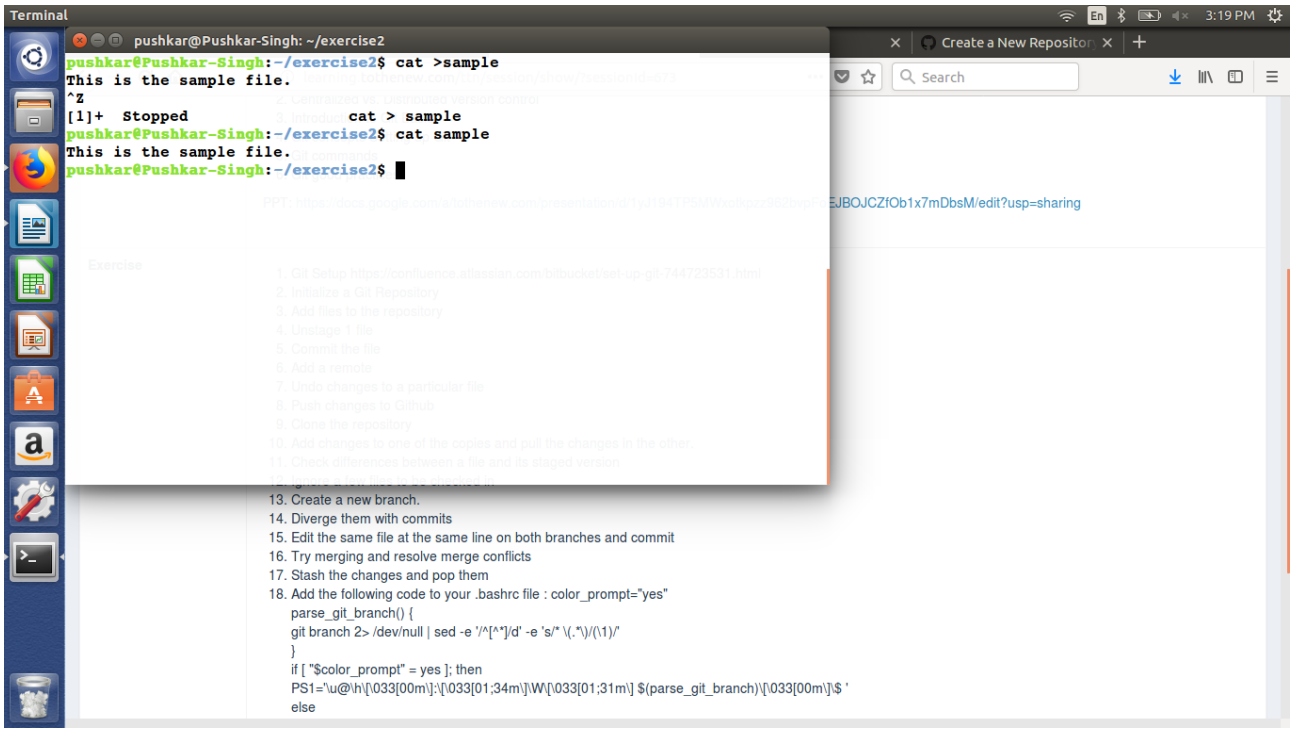
Q1. Git Setup https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html

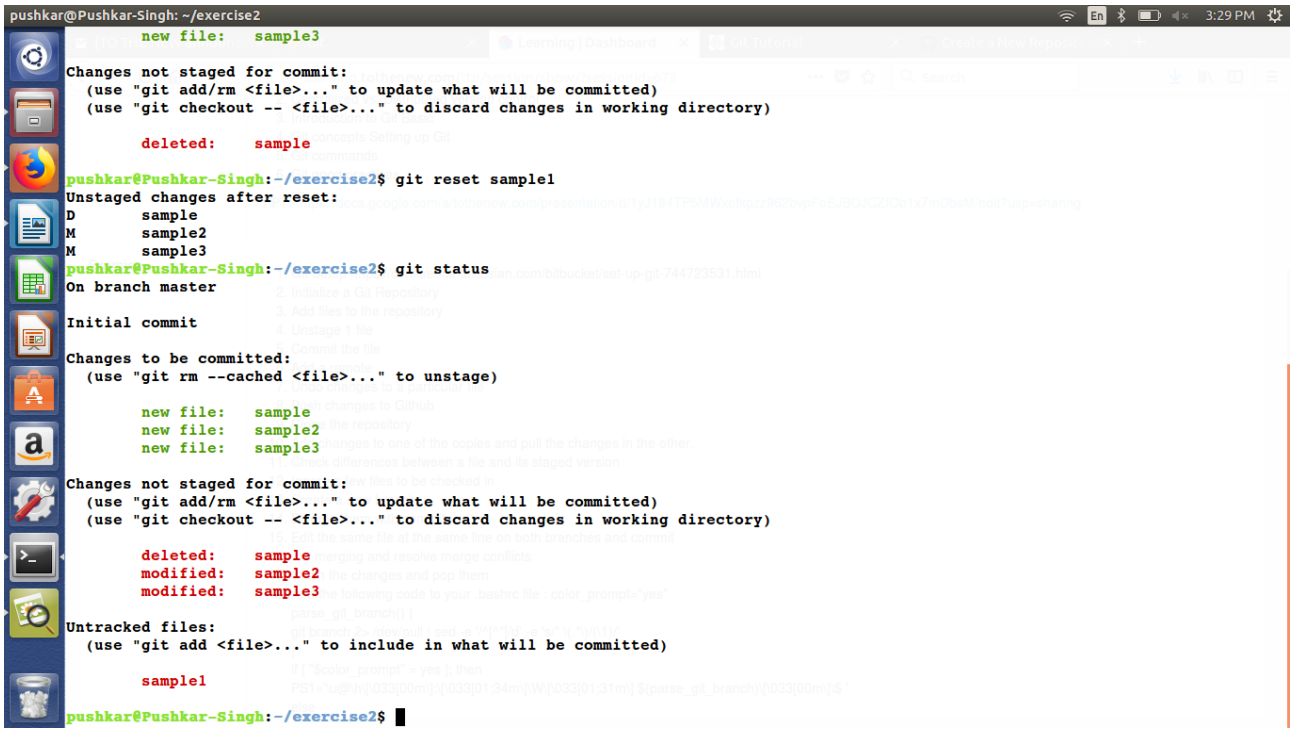
Q2. Initialize a Git Repository



Q3. Add files to the repository.



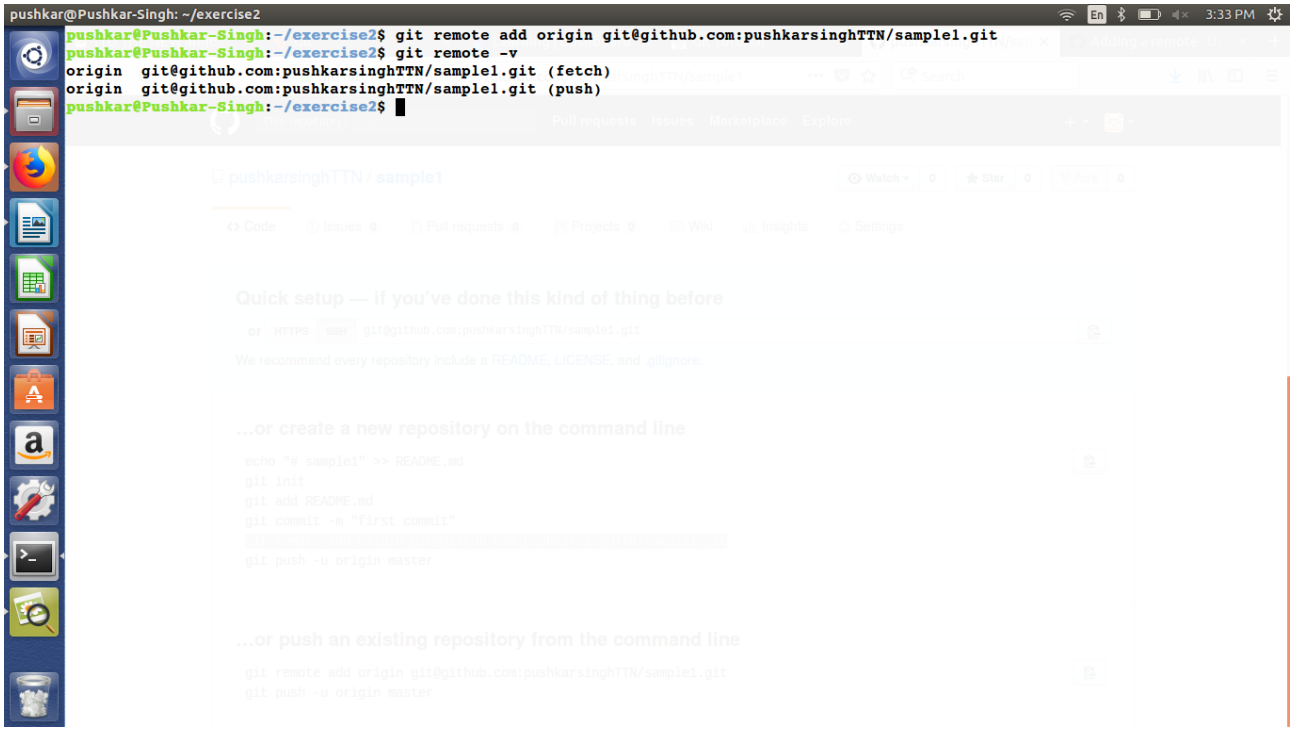
Q4. Unstage 1 file



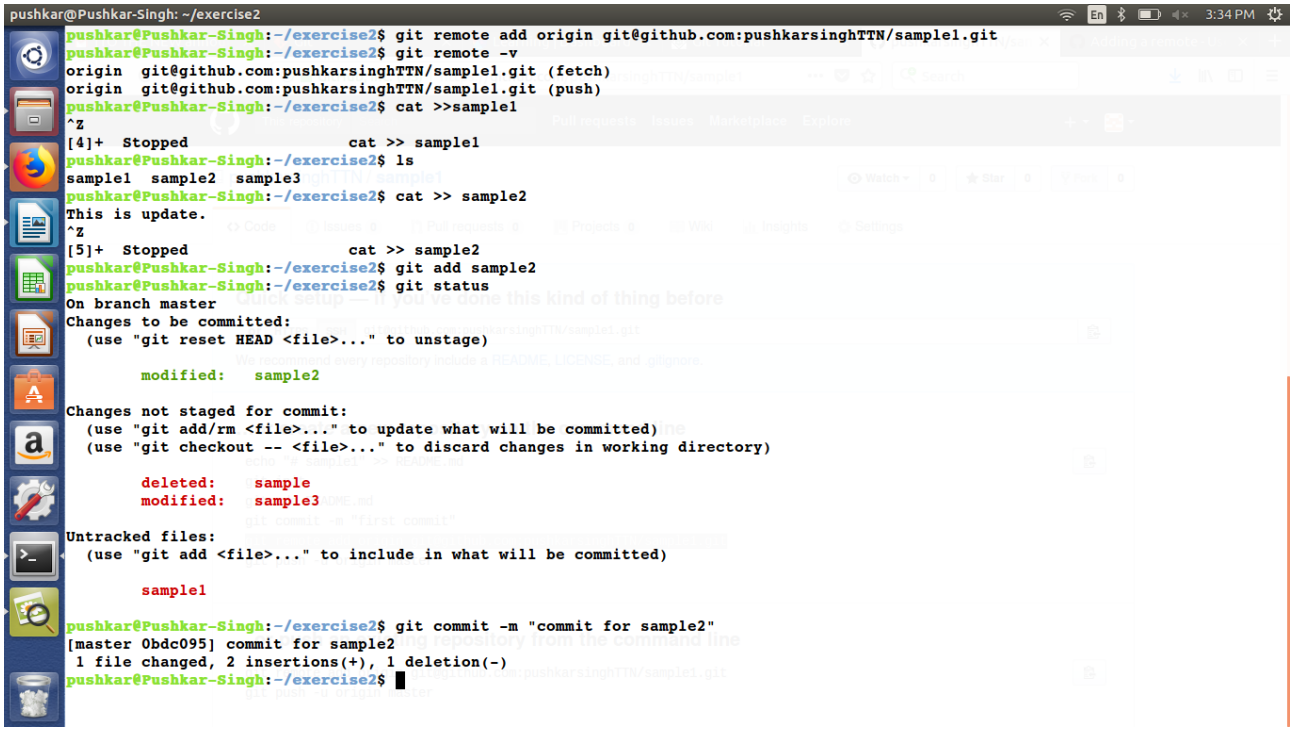
Q5. Commit the file.

```
pushkar@Pushkar-Singh:~/exercise2$ git commit
Aborting commit due to empty commit message.
pushkar@Pushkar-Singh:~/exercise2$ git commit -m "This is commit"
[master (root-commit) 8830b0a] This is commit
 3 files changed, 3 insertions(+)
 create mode 100644 sample
 create mode 100644 sample2
 create mode 100644 sample3
pushkar@Pushkar-Singh:~/exercise2$
```

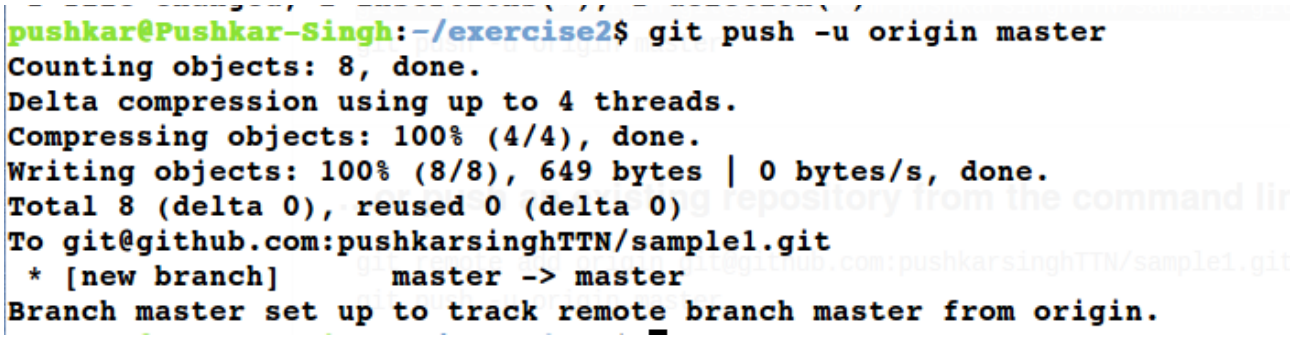
Q6. Add a remote.



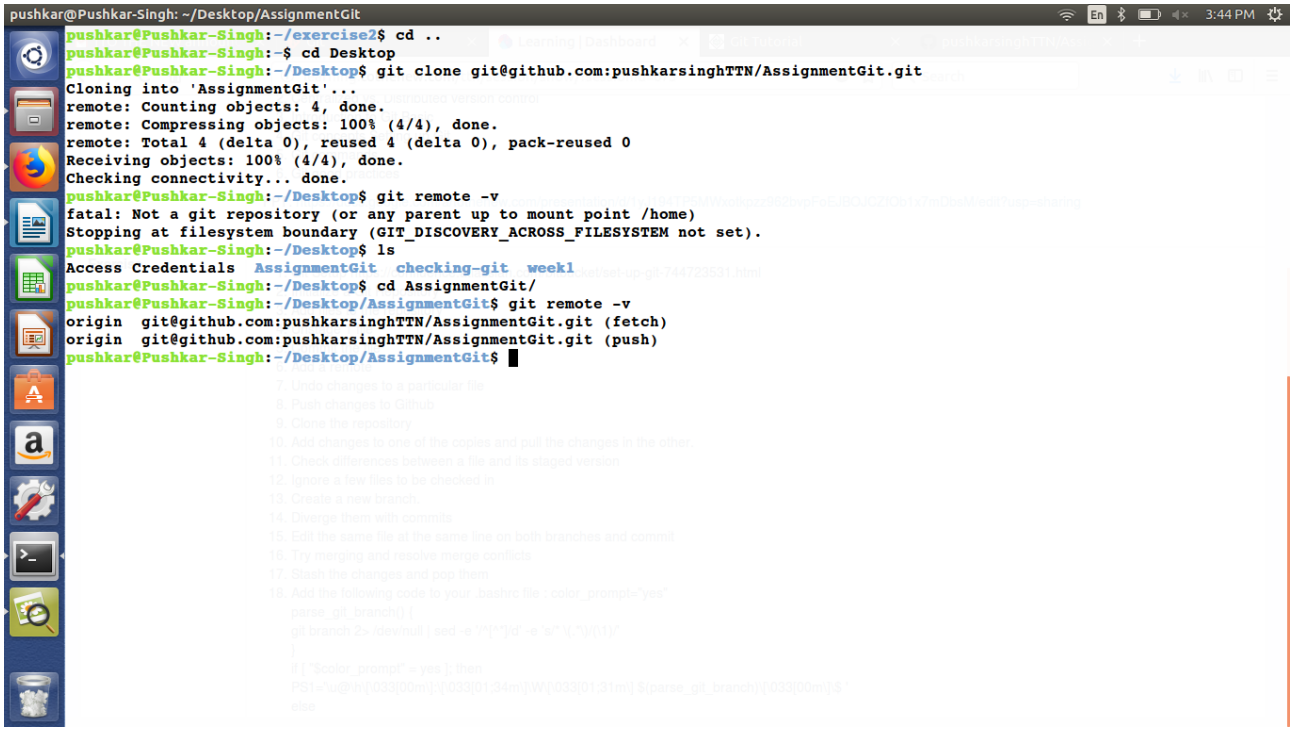
Q7. Undo changes to a particular file.



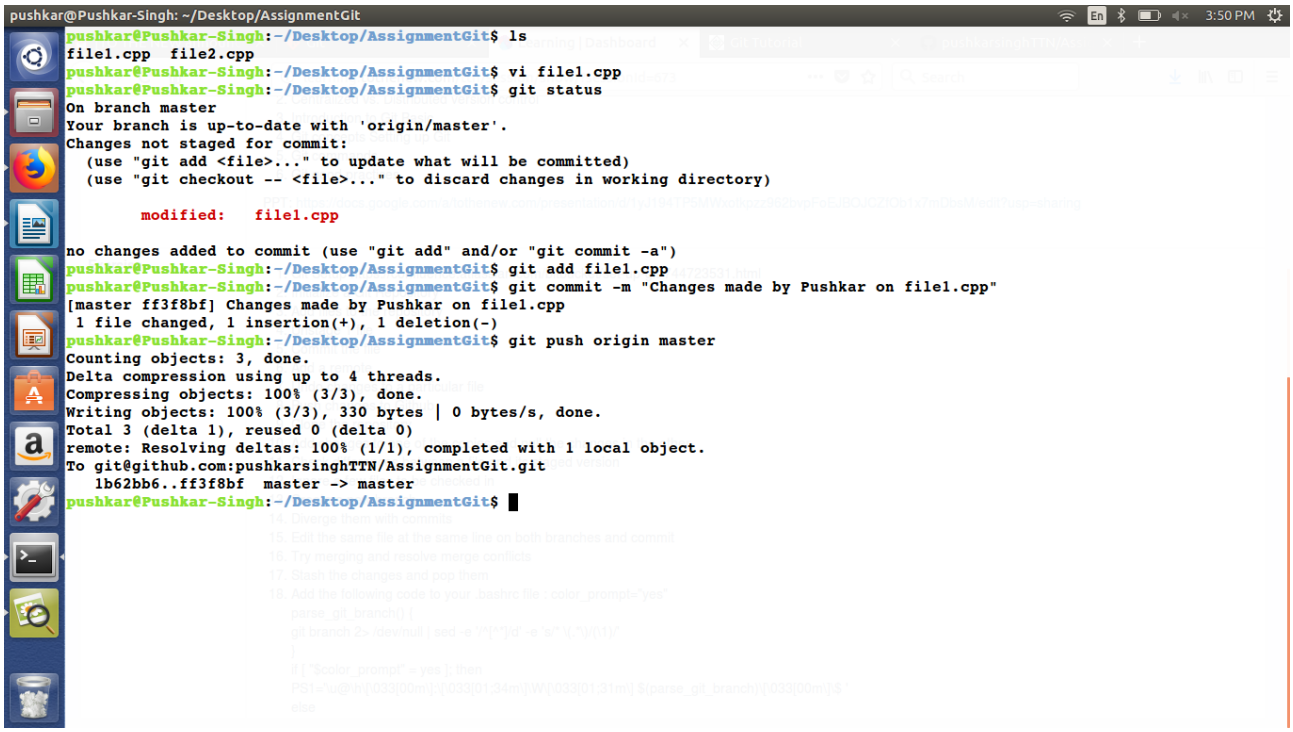
Q8. Push changes to Github.



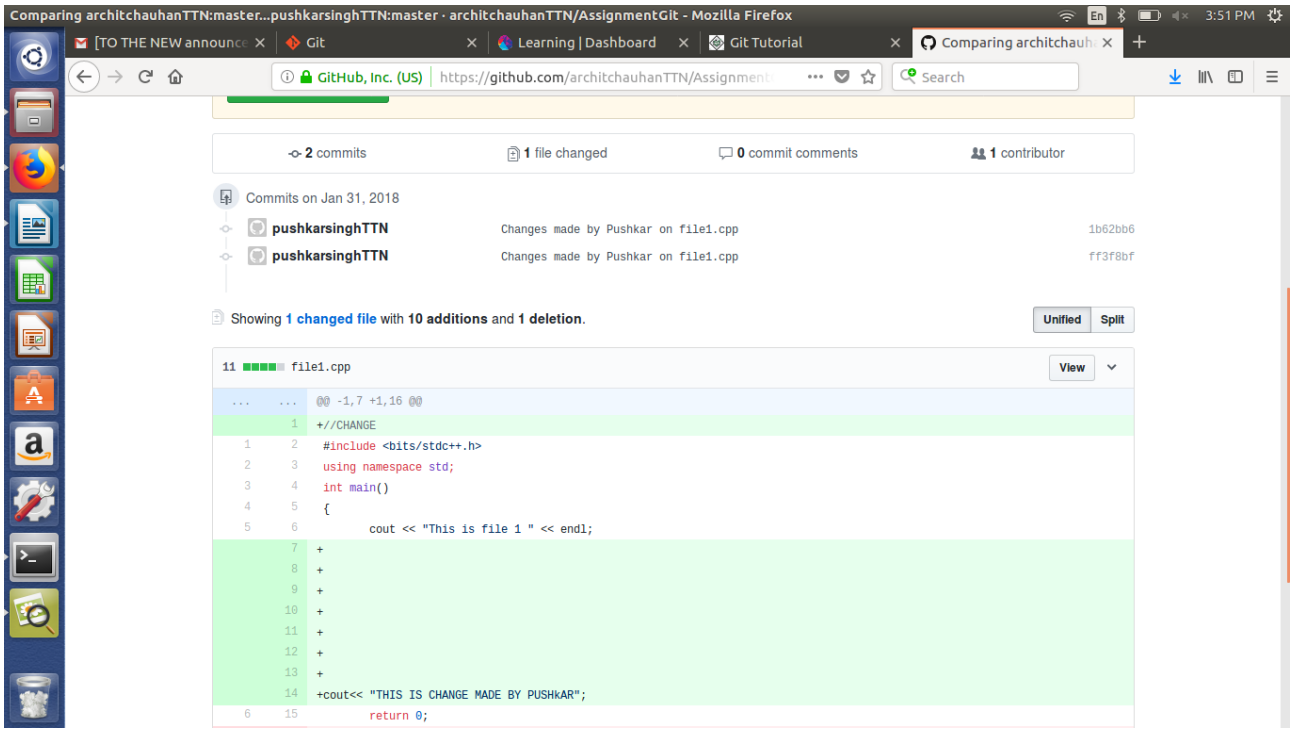
Q9. Clone the repository.



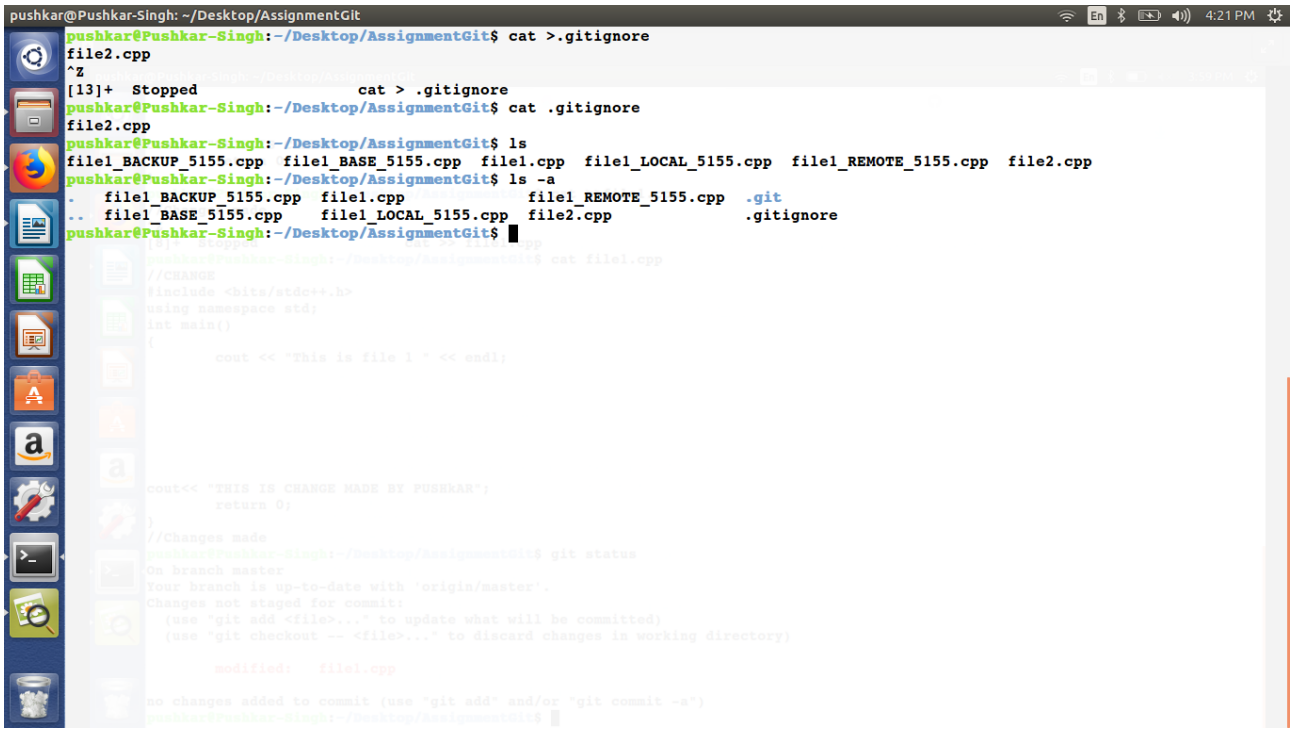
Q10. Add changes to one of the copies and pull the changes in the other.



Q11. Check differences between a file and its staged version



Q12. Ignore a few files to be checked in.

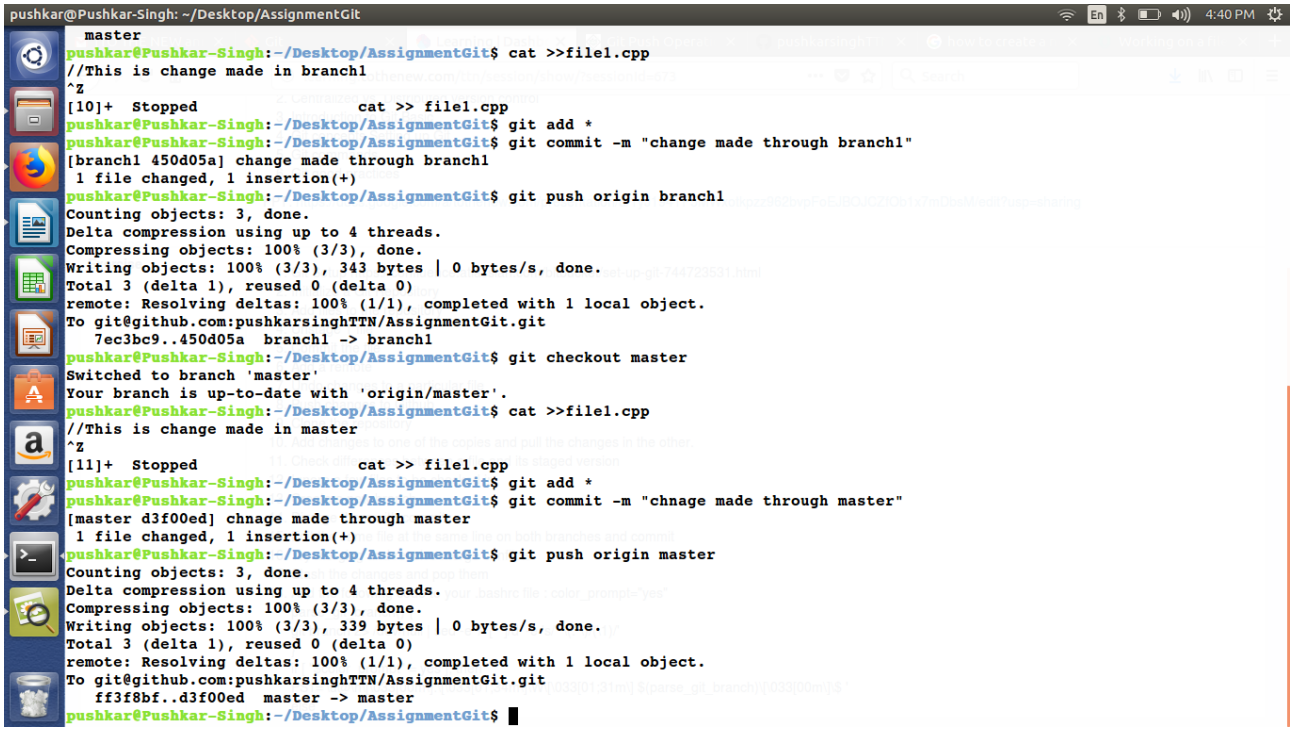


Q13. Create a new branch.

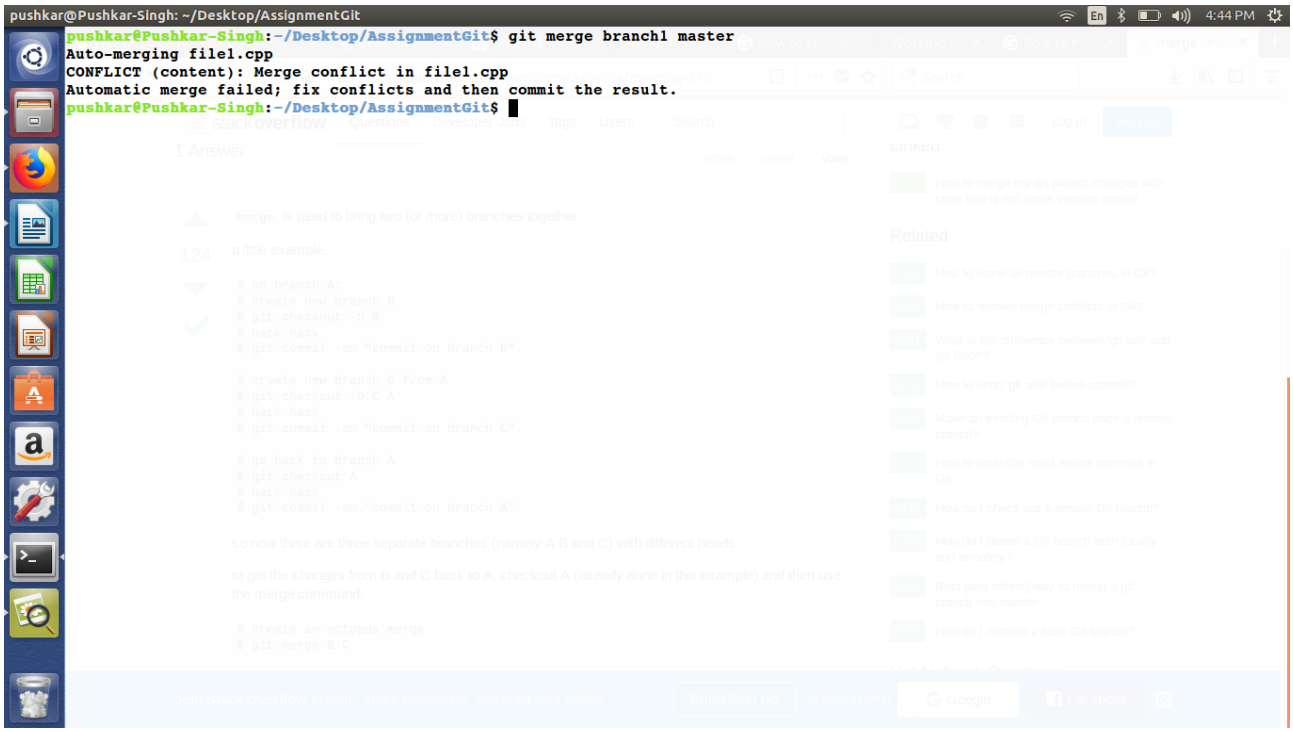
Q14. Diverge them with commits

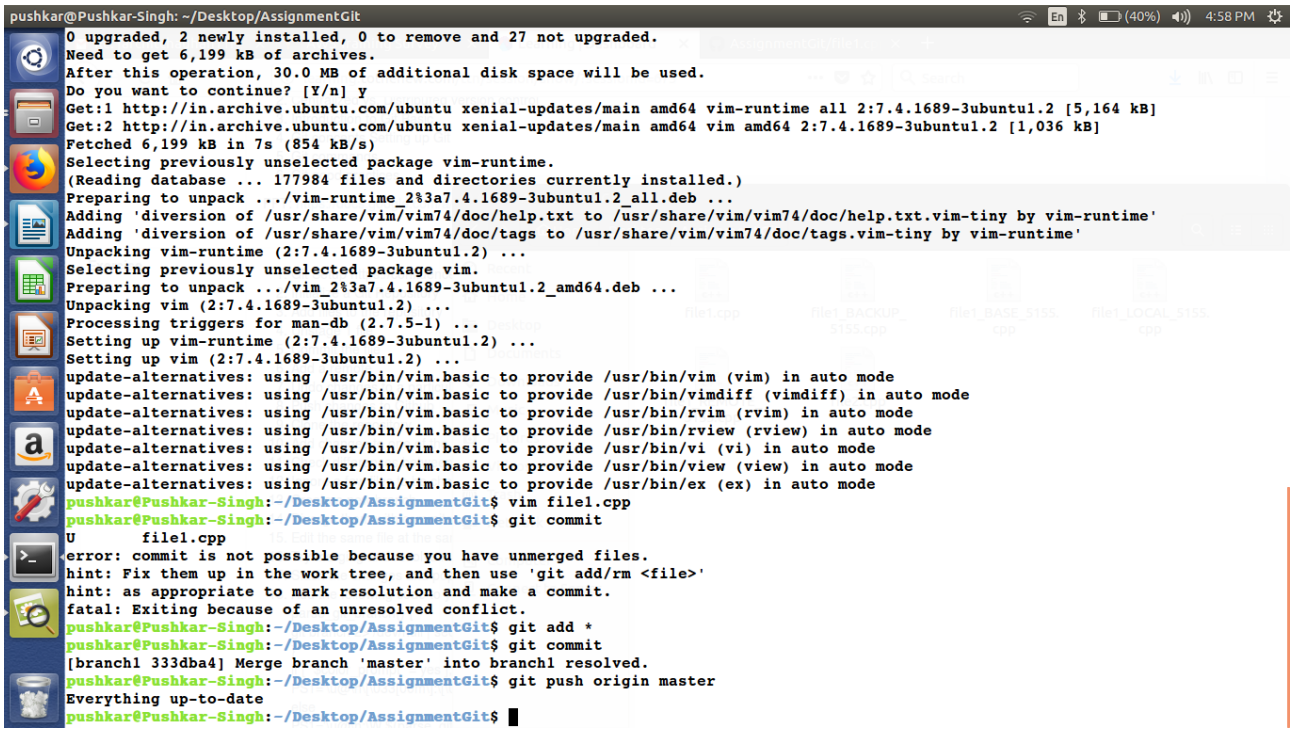
```
pushkar@Pushkar-Singh:~/Desktop/AssignmentGit$ cat >>file2.cpp
//CHANGES MADE TO FILE2 BY PUSHKAR
^Z
[9]+  Stopped                 cat >> file2.cpp
pushkar@Pushkar-Singh:~/Desktop/AssignmentGit$ git add *
pushkar@Pushkar-Singh:~/Desktop/AssignmentGit$ git commit -m "this is for new branch"
[branch1 7ec3bc9] this is for new branch
 2 files changed, 2 insertions(+), 1 deletion(-)
pushkar@Pushkar-Singh:~/Desktop/AssignmentGit$ git push origin branch1
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 396 bytes | 0 bytes/s, done.
Total 4 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To git@github.com:pushkarsinghTTN/AssignmentGit.git
 * [new branch]      branch1 -> branch1
```

Q15. Edit the same file at the same line on both branches and commit.



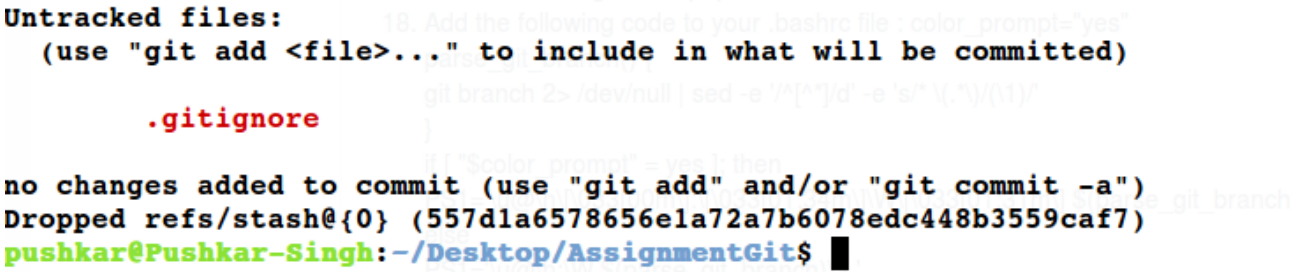
Q16. Try merging and resolve merge conflicts.





Q17. Stash the changes and pop them

```
pushkar@Pushkar-Singh:~/Desktop/AssignmentGit$ cat >>file1.cpp
//Changes2
^Z
[15]+  Stopped                 cat >> file1.cpp
pushkar@Pushkar-Singh:~/Desktop/AssignmentGit$ git status
On branch branch1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   file1.cpp

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore

no changes added to commit (use "git add" and/or "git commit -a")
pushkar@Pushkar-Singh:~/Desktop/AssignmentGit$ git stash
Saved working directory and index state WIP on branch1: 333dba4 Merge branch 'master' into branch1 resolved.
HEAD is now at 333dba4 Merge branch 'master' into branch1
pushkar@Pushkar-Singh:~/Desktop/AssignmentGit$ git status
On branch branch1
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore

nothing added to commit but untracked files present (use "git add" to track)
pushkar@Pushkar-Singh:~/Desktop/AssignmentGit$ git stash list
stash@{0}: WIP on branch1: 333dba4 Merge branch 'master' into branch1 resolved.
pushkar@Pushkar-Singh:~/Desktop/AssignmentGit$ git stash pop
On branch branch1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   file1.cpp
```



```
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (557d1a6578656e1a72a7b6078edc448b3559caf7)
pushkar@Pushkar-Singh:~/Desktop/AssignmentGit$
```

Q18. 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