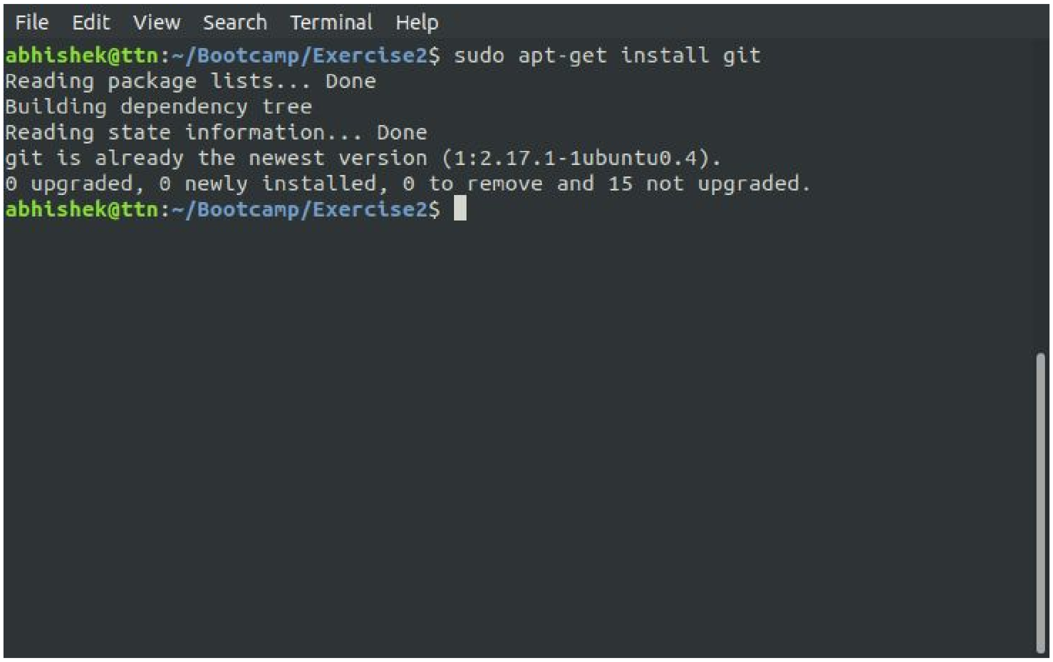
Version control & Git

Exercise 2

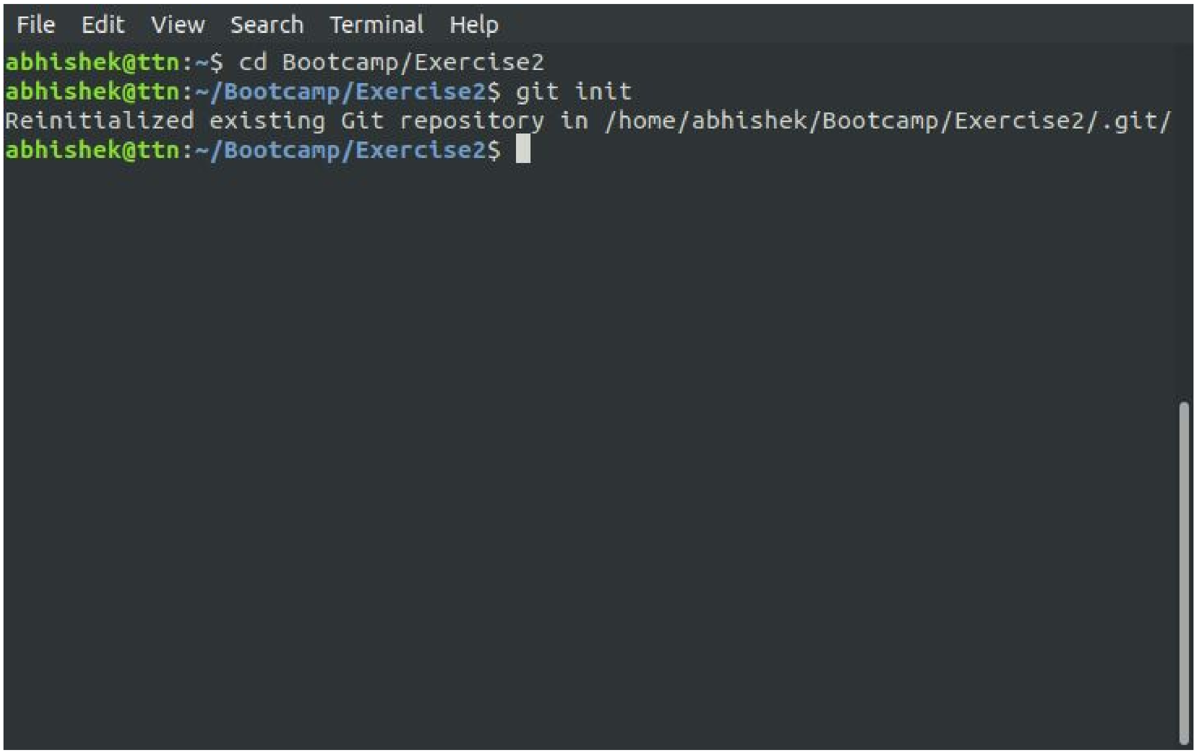
Abhishek Maurya

1. **Git Setup** https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html
    - `sudo apt-get install git`



```
abhishek@ttn:~/Bootcamp/Exercise2$ sudo apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.17.1-1ubuntu0.4).
0 upgraded, 0 newly installed, 0 to remove and 15 not upgraded.
abhishek@ttn:~/Bootcamp/Exercise2$
```

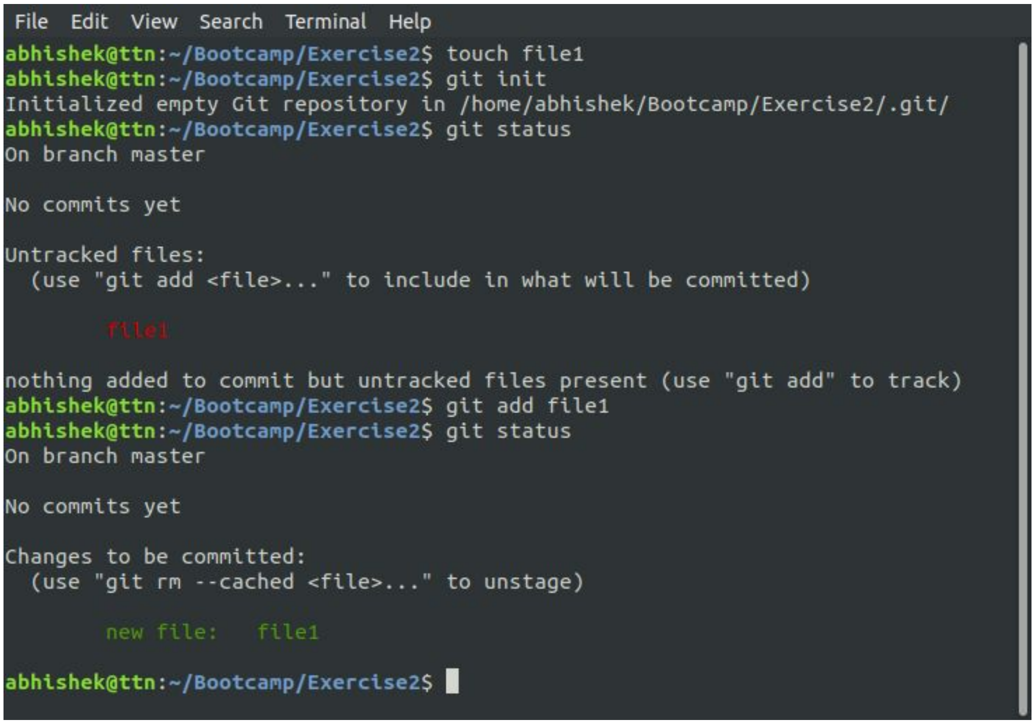
2. **Initialize a Git Repository**
    - `git init`



```
abhishek@ttn:~$ cd Bootcamp/Exercise2
abhishek@ttn:~/Bootcamp/Exercise2$ git init
Reinitialized existing Git repository in /home/abhishek/Bootcamp/Exercise2/.git/
abhishek@ttn:~/Bootcamp/Exercise2$
```

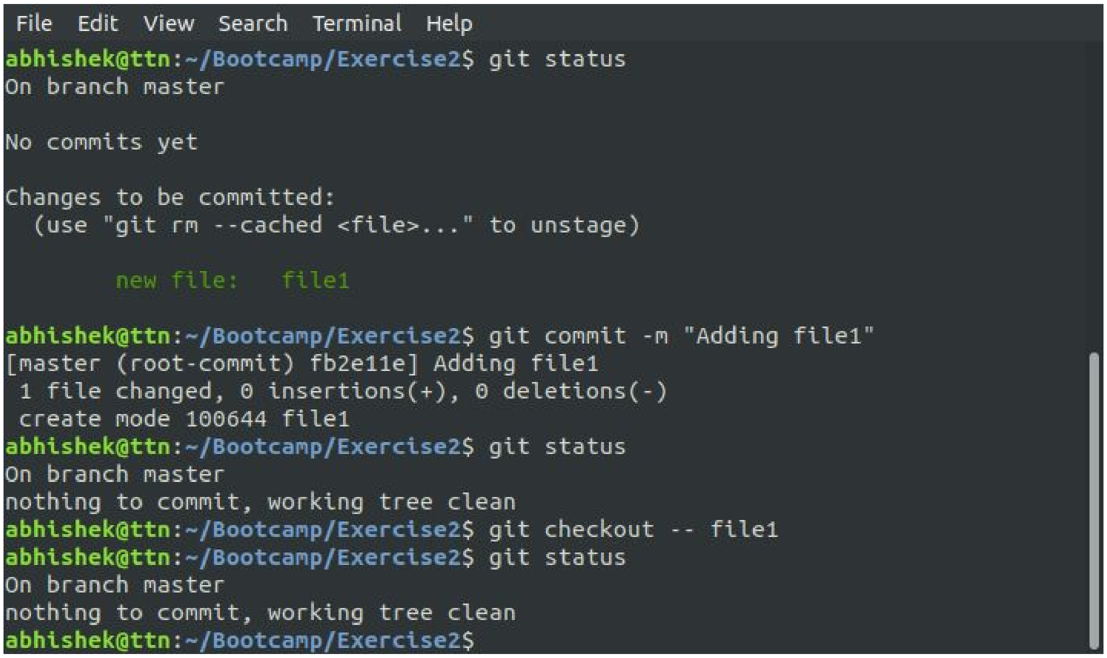
3. **Add files to the repository**
    - `touch file1`

      `git add file1`



```
abhishek@ttn:~/Bootcamp/Exercise2$ touch file1
abhishek@ttn:~/Bootcamp/Exercise2$ git init
Initialized empty Git repository in /home/abhishek/Bootcamp/Exercise2/.git/
abhishek@ttn:~/Bootcamp/Exercise2$ git status
On branch master

No commits yet

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

        file1

nothing added to commit but untracked files present (use "git add" to track)
abhishek@ttn:~/Bootcamp/Exercise2$ git add file1
abhishek@ttn:~/Bootcamp/Exercise2$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   file1

abhishek@ttn:~/Bootcamp/Exercise2$ 
```

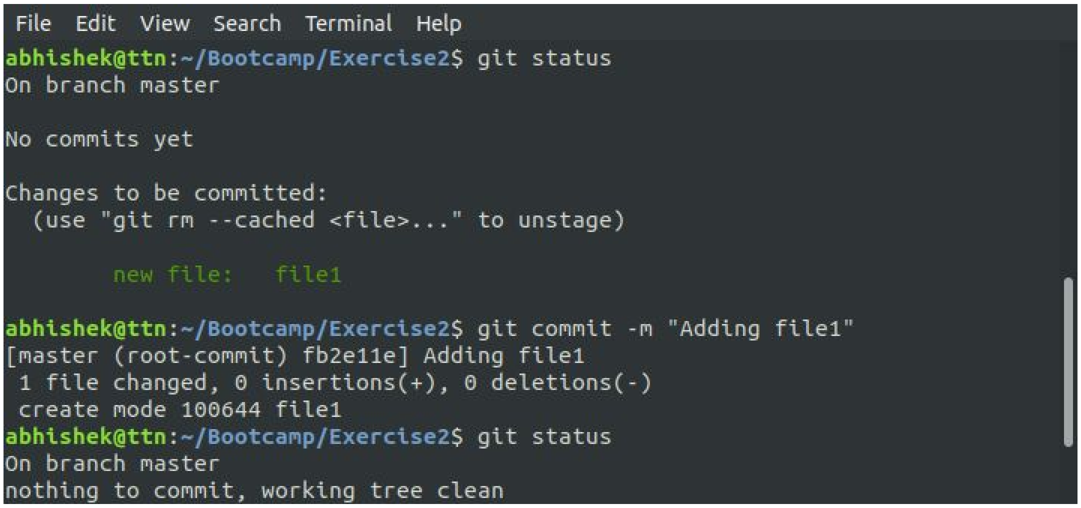
4. **Unstage 1 file**
    - `git reset HEAD file1`



```
abhishek@ttn:~/Bootcamp/Exercise2$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   file1

abhishek@ttn:~/Bootcamp/Exercise2$ git commit -m "Adding file1"
[master (root-commit) fb2e11e] Adding file1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file1
abhishek@ttn:~/Bootcamp/Exercise2$ git status
On branch master
nothing to commit, working tree clean
abhishek@ttn:~/Bootcamp/Exercise2$ git checkout -- file1
abhishek@ttn:~/Bootcamp/Exercise2$ git status
On branch master
nothing to commit, working tree clean
abhishek@ttn:~/Bootcamp/Exercise2$
```

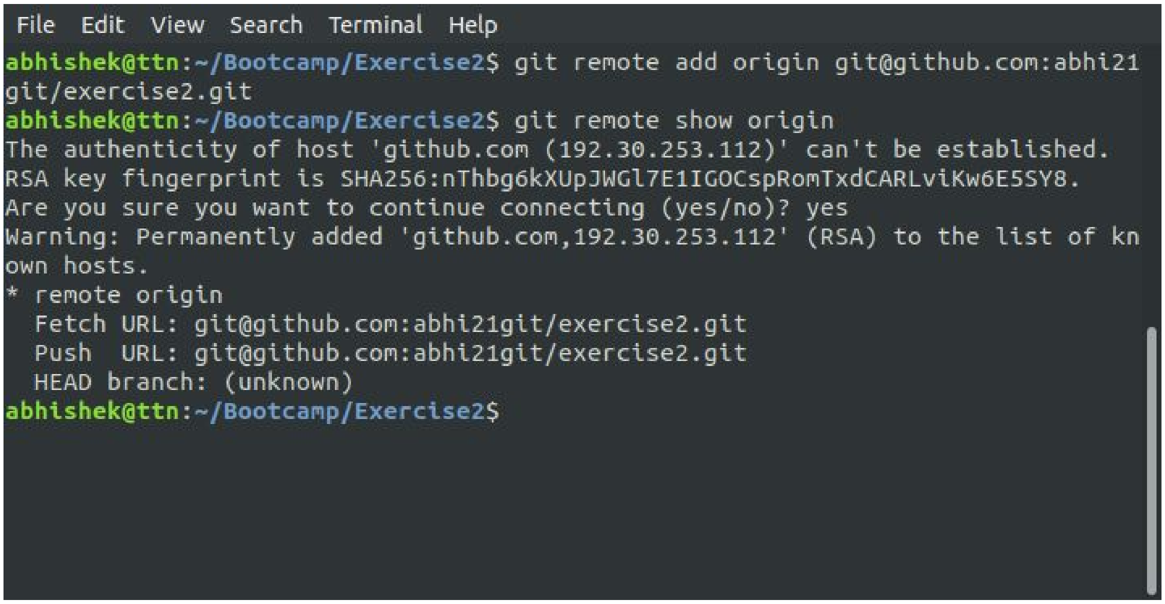
5. **Commit the file**

- git commit -m "Adding file1"



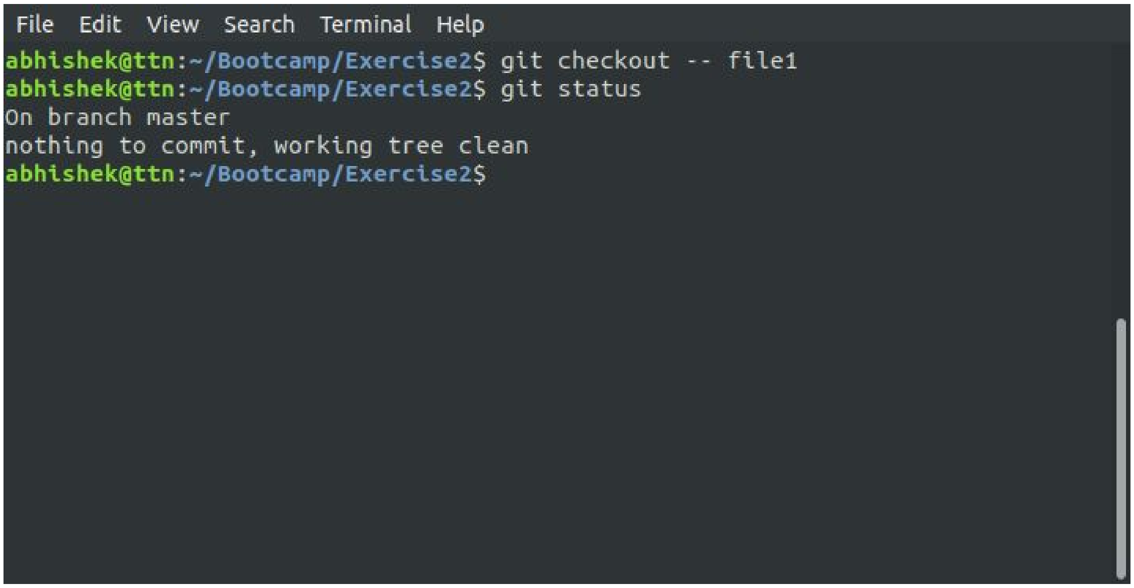
```
abhishek@ttn:~/Bootcamp/Exercise2$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   file1

abhishek@ttn:~/Bootcamp/Exercise2$ git commit -m "Adding file1"
[master (root-commit) fb2e11e] Adding file1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file1
abhishek@ttn:~/Bootcamp/Exercise2$ git status
On branch master
nothing to commit, working tree clean
```

6. **Add a remote**
   - git remote add origin git@github.com:abhi21git/exercise2.git



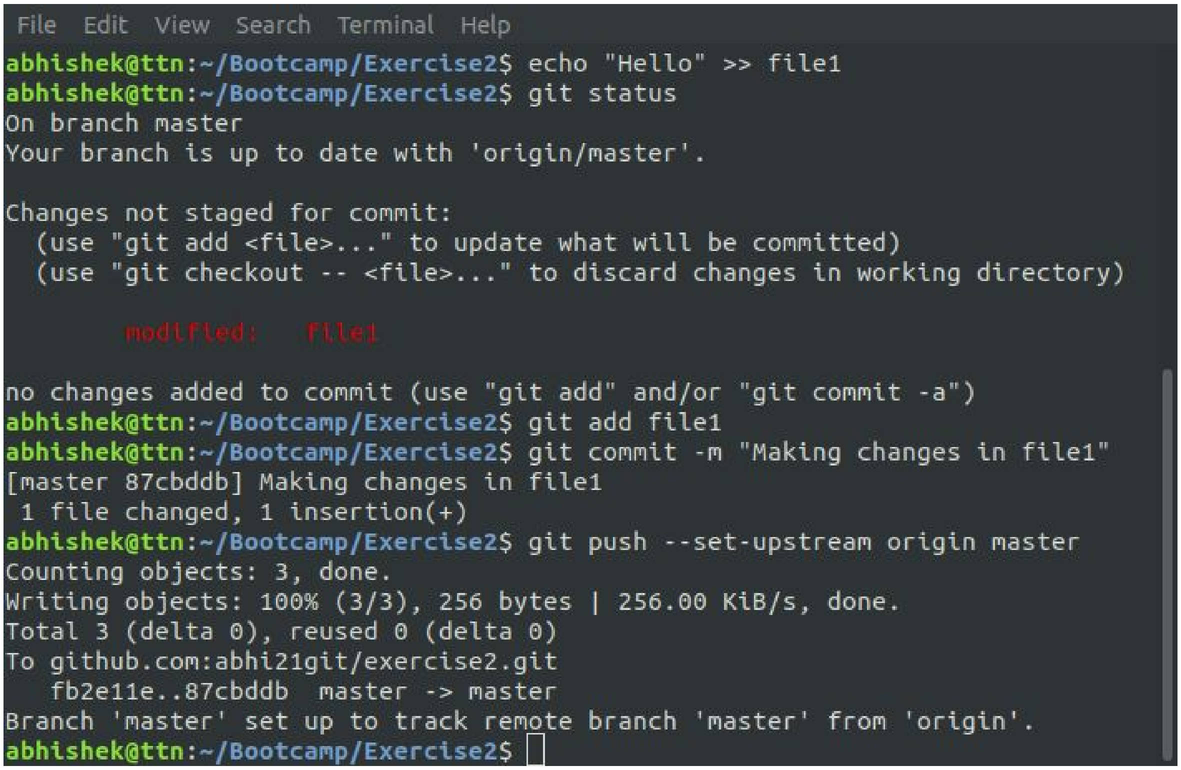
```
abhishek@ttn:~/Bootcamp/Exercise2$ git remote add origin git@github.com:abhi21
git/exercise2.git
abhishek@ttn:~/Bootcamp/Exercise2$ git remote show origin
The authenticity of host 'github.com (192.30.253.112)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.253.112' (RSA) to the list of kn
own hosts.
* remote origin
  Fetch URL: git@github.com:abhi21git/exercise2.git
  Push  URL: git@github.com:abhi21git/exercise2.git
  HEAD branch: (unknown)
abhishek@ttn:~/Bootcamp/Exercise2$
```

7. **Undo changes to a particular file**
   - git checkout -- file1



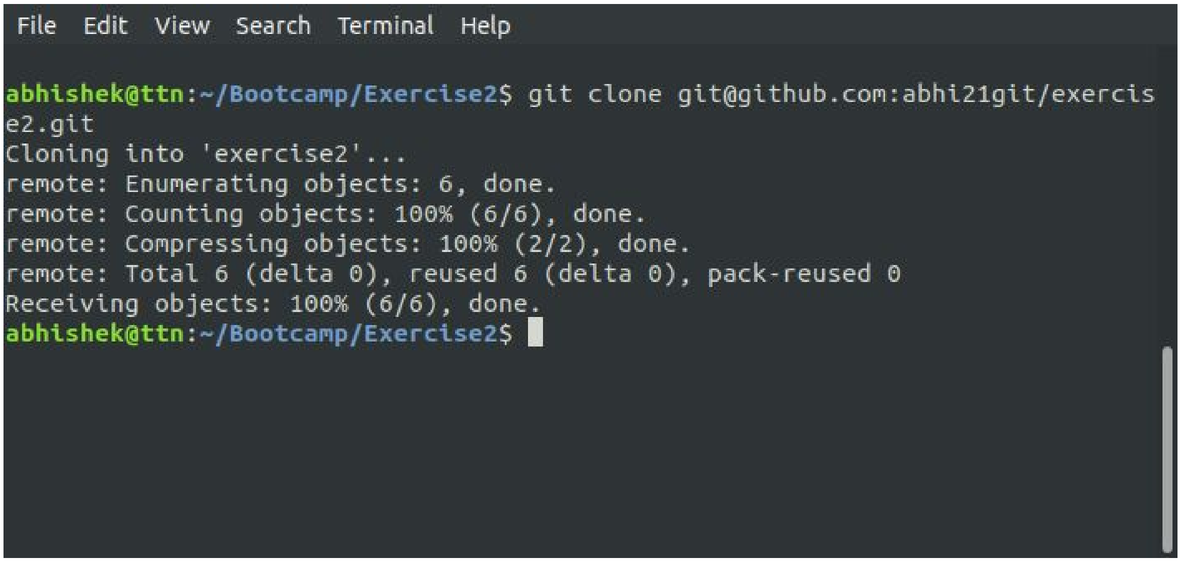
```
abhishek@ttn:~/Bootcamp/Exercise2$ git checkout -- file1
abhishek@ttn:~/Bootcamp/Exercise2$ git status
On branch master
nothing to commit, working tree clean
abhishek@ttn:~/Bootcamp/Exercise2$
```

8. **Push changes to Github**
    - `git push --set-upstream origin master`



```
abhishek@ttn:~/Bootcamp/Exercise2$ echo "Hello" >> file1
abhishek@ttn:~/Bootcamp/Exercise2$ git status
On branch master
Your branch is up to date with 'origin/master'.

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

no changes added to commit (use "git add" and/or "git commit -a")
abhishek@ttn:~/Bootcamp/Exercise2$ git add file1
abhishek@ttn:~/Bootcamp/Exercise2$ git commit -m "Making changes in file1"
[master 87cbddb] Making changes in file1
 1 file changed, 1 insertion(+)
abhishek@ttn:~/Bootcamp/Exercise2$ git push --set-upstream origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 256 bytes | 256.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:abhi21git/exercise2.git
   fb2e11e..87cbddb  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
abhishek@ttn:~/Bootcamp/Exercise2$
```

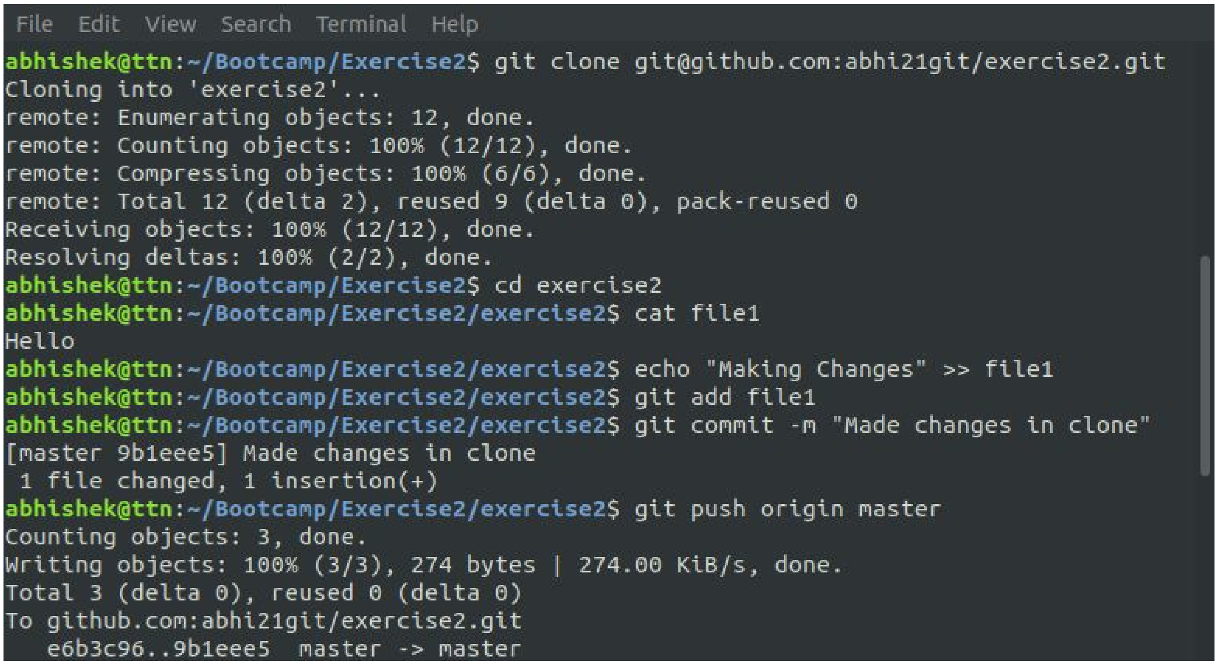
9. **Clone the repository**
    - `git clone git@github.com:abhi21git/exercise2.git`



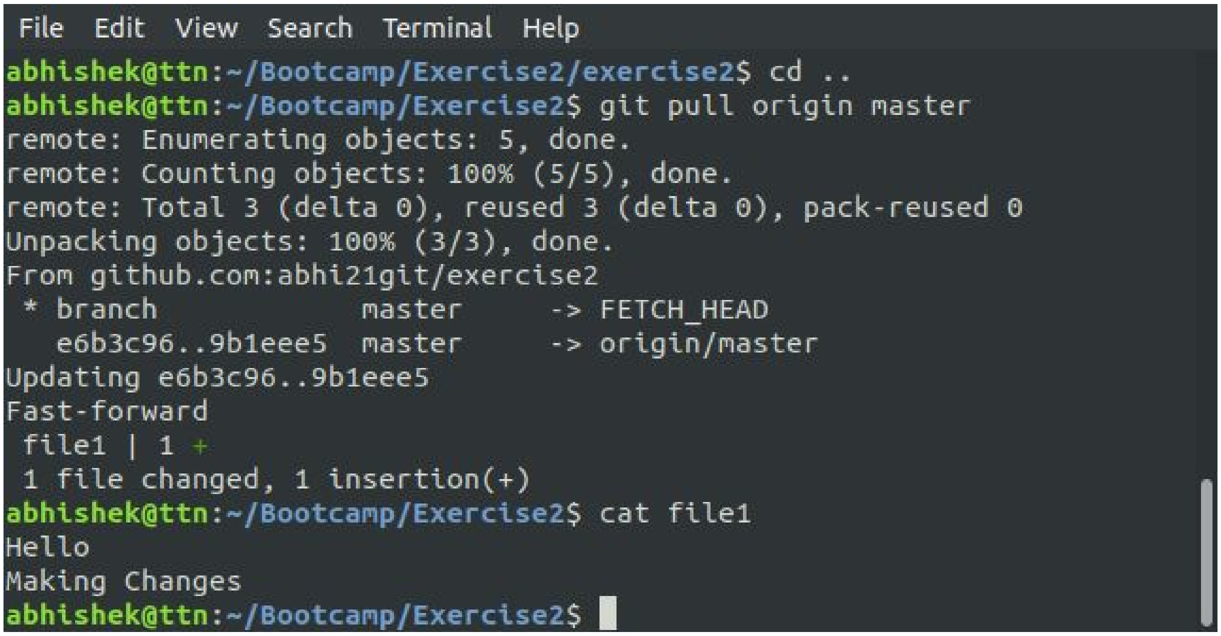
```
abhishek@ttn:~/Bootcamp/Exercise2$ git clone git@github.com:abhi21git/exercis
e2.git
Cloning into 'exercise2'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
abhishek@ttn:~/Bootcamp/Exercise2$ 
```

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

- `echo "Making changes" >> exercise2/file1 //at clone directory`
- `git pull origin master                  //at original directory`



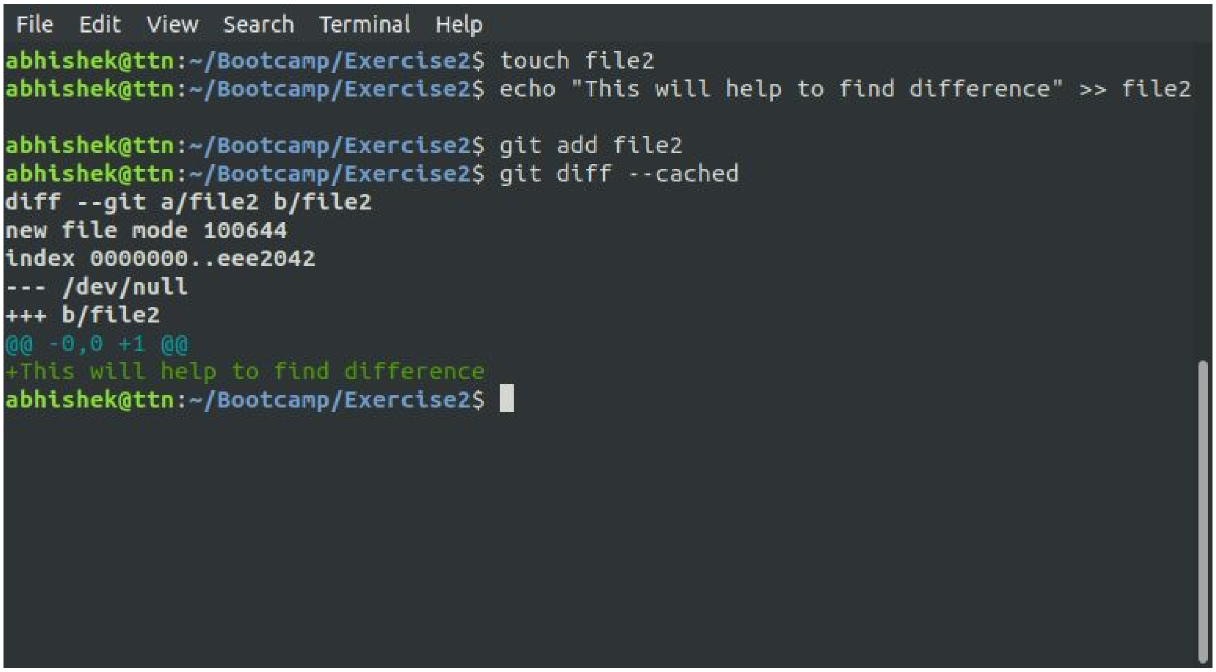
```
abhishek@ttn:~/Bootcamp/Exercise2$ git clone git@github.com:abhi21git/exercise2.git
Cloning into 'exercise2'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 12 (delta 2), reused 9 (delta 0), pack-reused 0
Receiving objects: 100% (12/12), done.
Resolving deltas: 100% (2/2), done.
abhishek@ttn:~/Bootcamp/Exercise2$ cd exercise2
abhishek@ttn:~/Bootcamp/Exercise2/exercise2$ cat file1
Hello
abhishek@ttn:~/Bootcamp/Exercise2/exercise2$ echo "Making Changes" >> file1
abhishek@ttn:~/Bootcamp/Exercise2/exercise2$ git add file1
abhishek@ttn:~/Bootcamp/Exercise2/exercise2$ git commit -m "Made changes in clone"
[master 9b1eee5] Made changes in clone
 1 file changed, 1 insertion(+)
abhishek@ttn:~/Bootcamp/Exercise2/exercise2$ git push origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 274 bytes | 274.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:abhi21git/exercise2.git
   e6b3c96..9b1eee5  master -> master
```



```
abhishek@ttn:~/Bootcamp/Exercise2/exercise2$ cd ..
abhishek@ttn:~/Bootcamp/Exercise2$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:abhi21git/exercise2
 * branch            master     -> FETCH_HEAD
   e6b3c96..9b1eee5  master     -> origin/master
Updating e6b3c96..9b1eee5
Fast-forward
 file1 | 1 +
 1 file changed, 1 insertion(+)
abhishek@ttn:~/Bootcamp/Exercise2$ cat file1
Hello
Making Changes
abhishek@ttn:~/Bootcamp/Exercise2$ 
```

11. Check differences between a file and its staged version

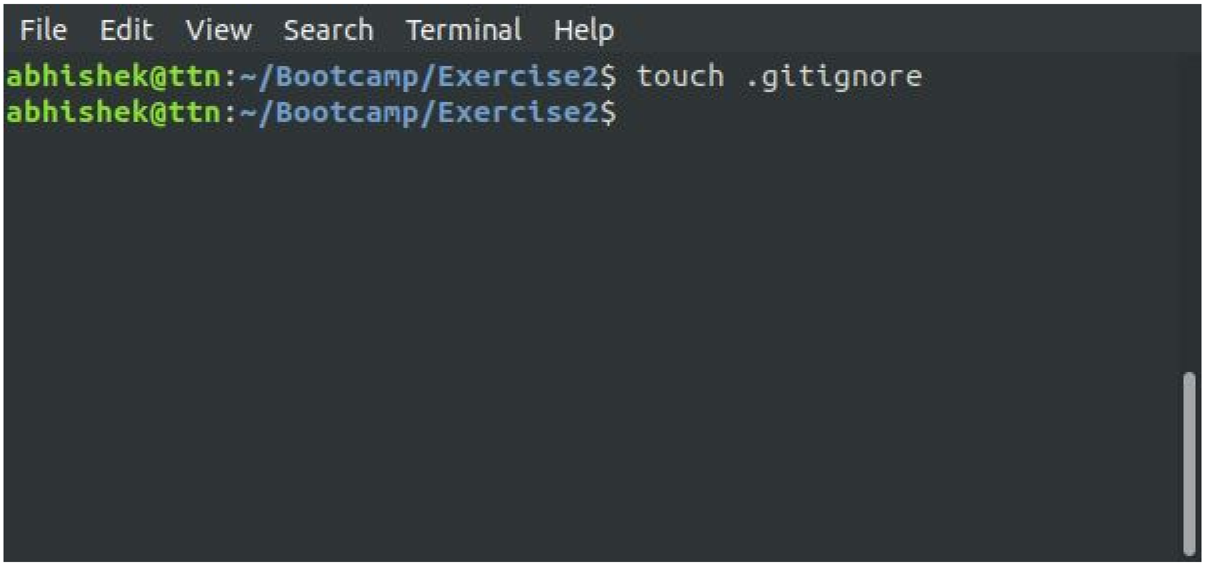
- `git diff --cached`



```
abhishek@ttn:~/Bootcamp/Exercise2$ touch file2
abhishek@ttn:~/Bootcamp/Exercise2$ echo "This will help to find difference" >> file2

abhishek@ttn:~/Bootcamp/Exercise2$ git add file2
abhishek@ttn:~/Bootcamp/Exercise2$ git diff --cached
diff --git a/file2 b/file2
new file mode 100644
index 0000000..eee2042
--- /dev/null
+++ b/file2
@@ -0,0 +1 @@
+This will help to find difference
abhishek@ttn:~/Bootcamp/Exercise2$ 
```

12. Ignore a few files to be checked in

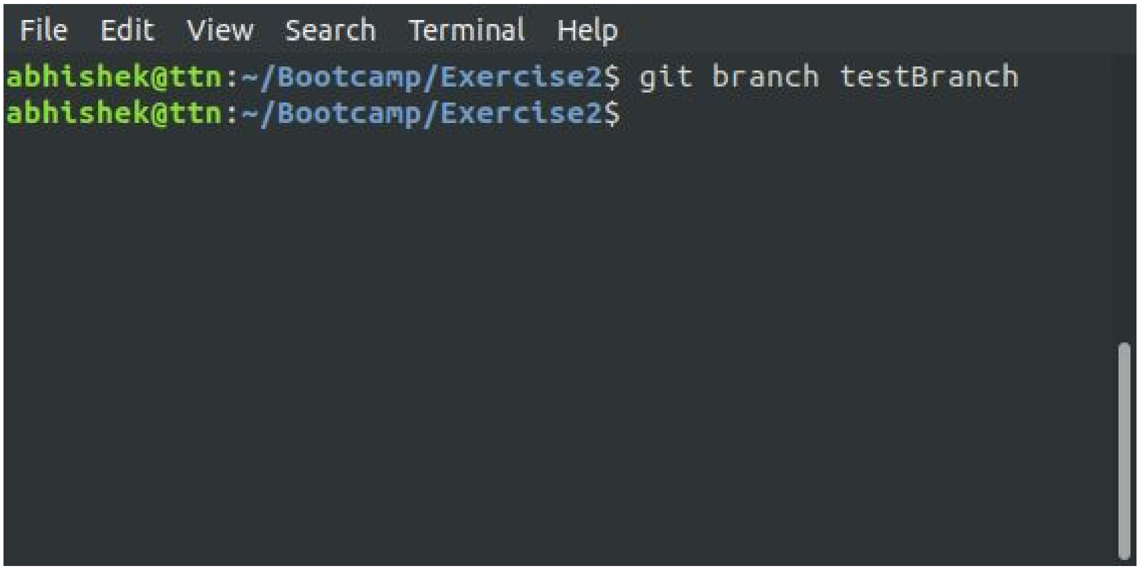
- `touch .gitignore`



```
abhishek@ttn:~/Bootcamp/Exercise2$ touch .gitignore
abhishek@ttn:~/Bootcamp/Exercise2$
```

13. Create a new branch.

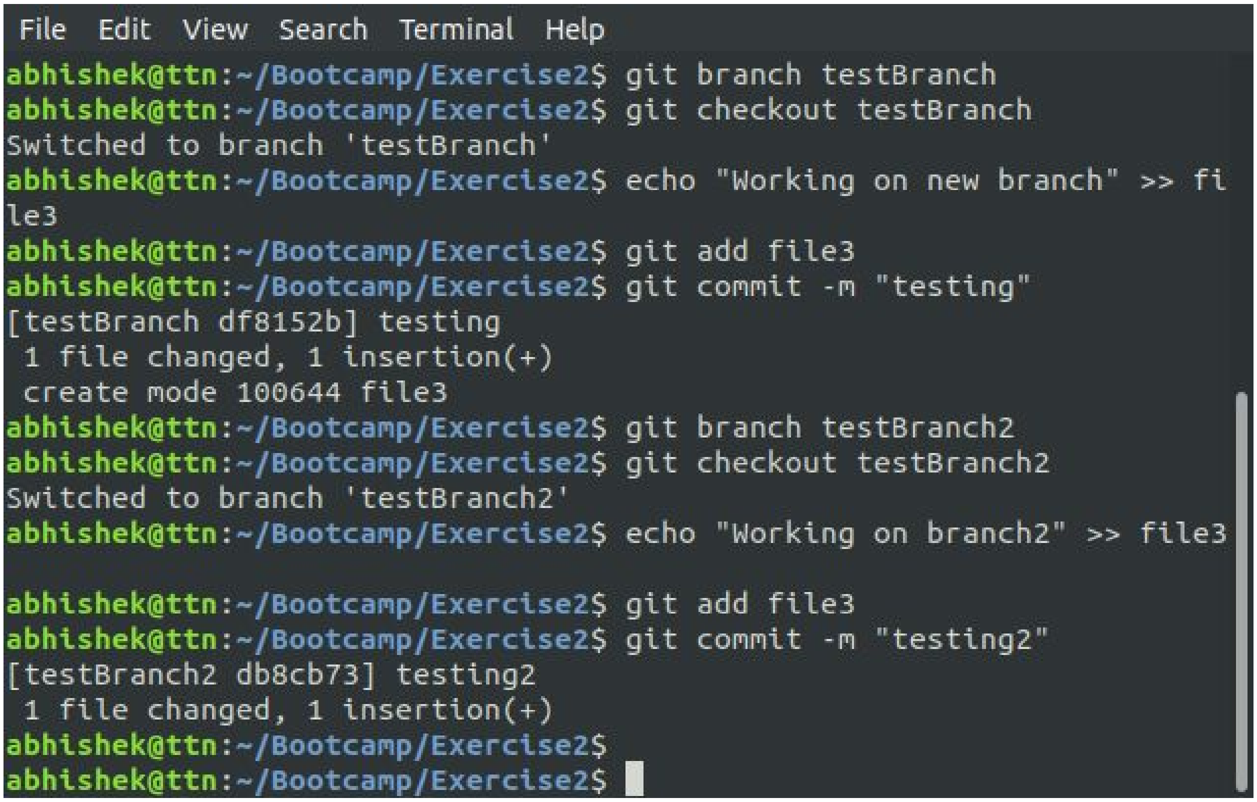
- `git branch testBranch`



```
abhishek@ttn:~/Bootcamp/Exercise2$ git branch testBranch
abhishek@ttn:~/Bootcamp/Exercise2$
```

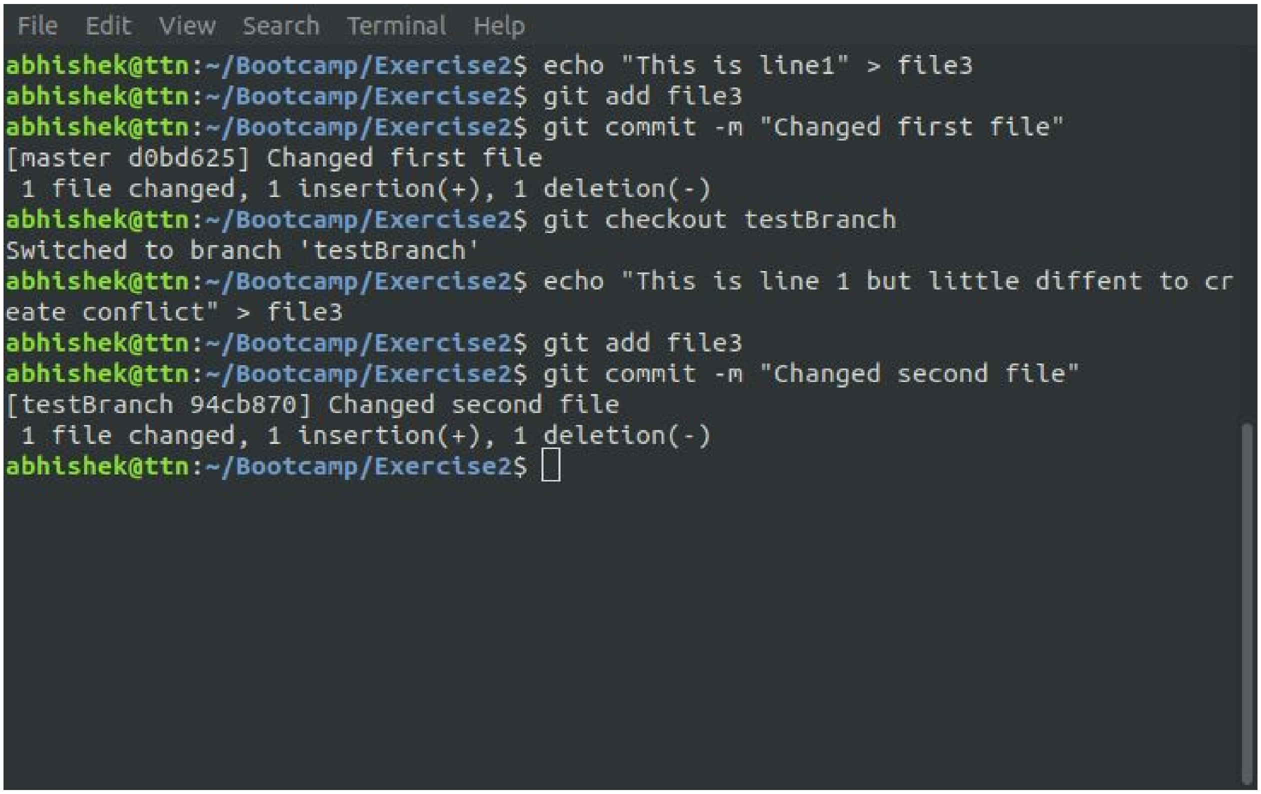
14. Diverge them with commits

-
  ```
  git branch testBranch

  git checkout testBranch

  echo "Working on test branch" >> file3

  git add file3

  git commit -m "testing"
  ```



```
abhishek@ttn:~/Bootcamp/Exercise2$ git branch testBranch
abhishek@ttn:~/Bootcamp/Exercise2$ git checkout testBranch
Switched to branch 'testBranch'
abhishek@ttn:~/Bootcamp/Exercise2$ echo "Working on new branch" >> fi
le3
abhishek@ttn:~/Bootcamp/Exercise2$ git add file3
abhishek@ttn:~/Bootcamp/Exercise2$ git commit -m "testing"
[testBranch df8152b] testing
 1 file changed, 1 insertion(+)
 create mode 100644 file3
abhishek@ttn:~/Bootcamp/Exercise2$ git branch testBranch2
abhishek@ttn:~/Bootcamp/Exercise2$ git checkout testBranch2
Switched to branch 'testBranch2'
abhishek@ttn:~/Bootcamp/Exercise2$ echo "Working on branch2" >> file3

abhishek@ttn:~/Bootcamp/Exercise2$ git add file3
abhishek@ttn:~/Bootcamp/Exercise2$ git commit -m "testing2"
[testBranch2 db8cb73] testing2
 1 file changed, 1 insertion(+)
abhishek@ttn:~/Bootcamp/Exercise2$
abhishek@ttn:~/Bootcamp/Exercise2$
```

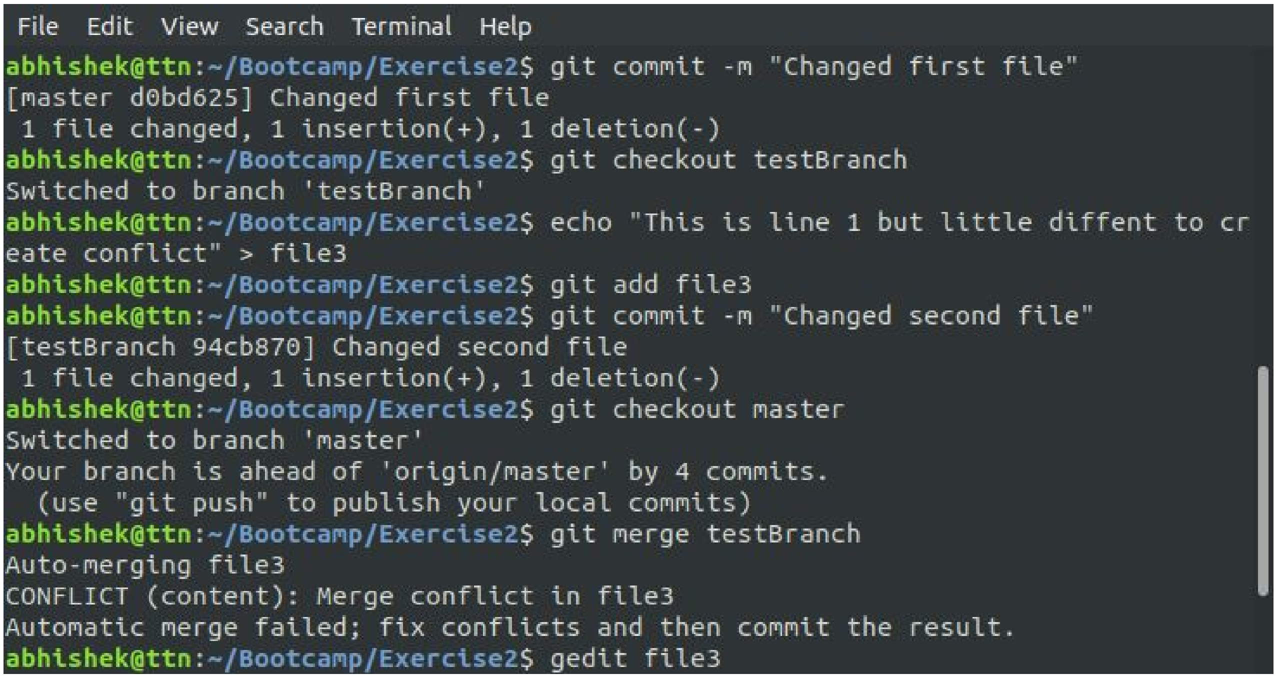
15. Edit the same file at the same line on both branches and commit



```
abhishek@ttn:~/Bootcamp/Exercise2$ echo "This is line1" > file3
abhishek@ttn:~/Bootcamp/Exercise2$ git add file3
abhishek@ttn:~/Bootcamp/Exercise2$ git commit -m "Changed first file"
[master d0bd625] Changed first file
 1 file changed, 1 insertion(+), 1 deletion(-)
abhishek@ttn:~/Bootcamp/Exercise2$ git checkout testBranch
Switched to branch 'testBranch'
abhishek@ttn:~/Bootcamp/Exercise2$ echo "This is line 1 but little diffent to cr
eate conflict" > file3
abhishek@ttn:~/Bootcamp/Exercise2$ git add file3
abhishek@ttn:~/Bootcamp/Exercise2$ git commit -m "Changed second file"
[testBranch 94cb870] Changed second file
 1 file changed, 1 insertion(+), 1 deletion(-)
abhishek@ttn:~/Bootcamp/Exercise2$
```

16. Try merging and resolve merge conflicts

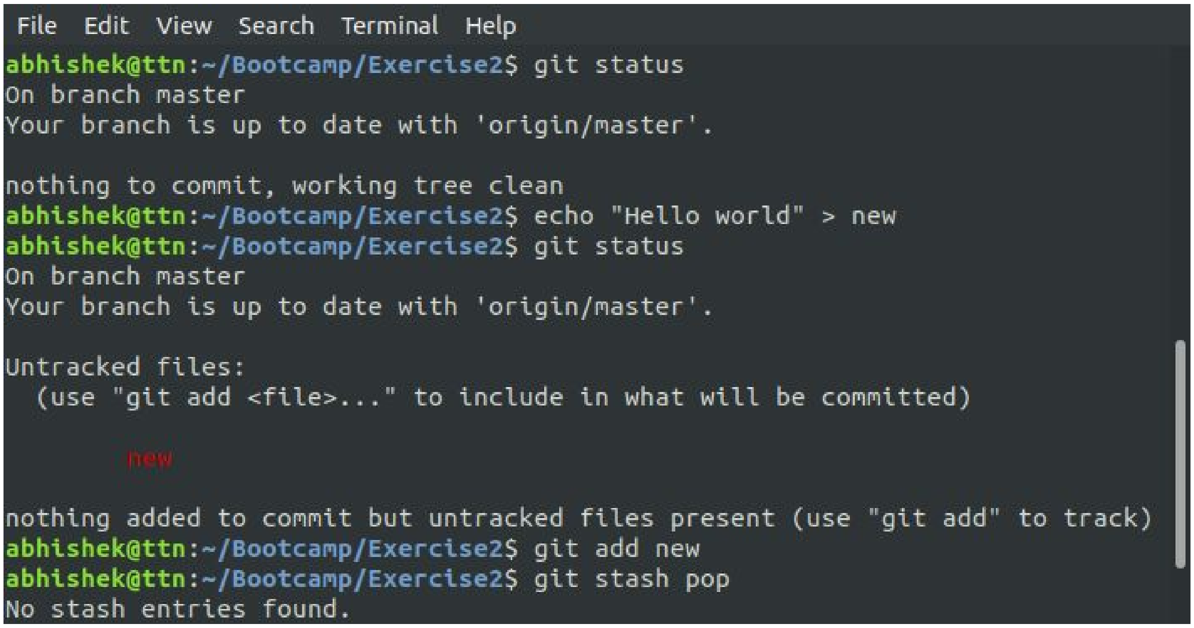
- `git merge testBranch`



```
abhishek@ttn:~/Bootcamp/Exercise2$ git commit -m "Changed first file"
[master d0bd625] Changed first file
 1 file changed, 1 insertion(+), 1 deletion(-)
abhishek@ttn:~/Bootcamp/Exercise2$ git checkout testBranch
Switched to branch 'testBranch'
abhishek@ttn:~/Bootcamp/Exercise2$ echo "This is line 1 but little diffent to cr
eate conflict" > file3
abhishek@ttn:~/Bootcamp/Exercise2$ git add file3
abhishek@ttn:~/Bootcamp/Exercise2$ git commit -m "Changed second file"
[testBranch 94cb870] Changed second file
 1 file changed, 1 insertion(+), 1 deletion(-)
abhishek@ttn:~/Bootcamp/Exercise2$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 4 commits.
  (use "git push" to publish your local commits)
abhishek@ttn:~/Bootcamp/Exercise2$ git merge testBranch
Auto-merging file3
CONFLICT (content): Merge conflict in file3
Automatic merge failed; fix conflicts and then commit the result.
abhishek@ttn:~/Bootcamp/Exercise2$ gedit file3
```

17. Stash the changes and pop them

- `git  stash pop`

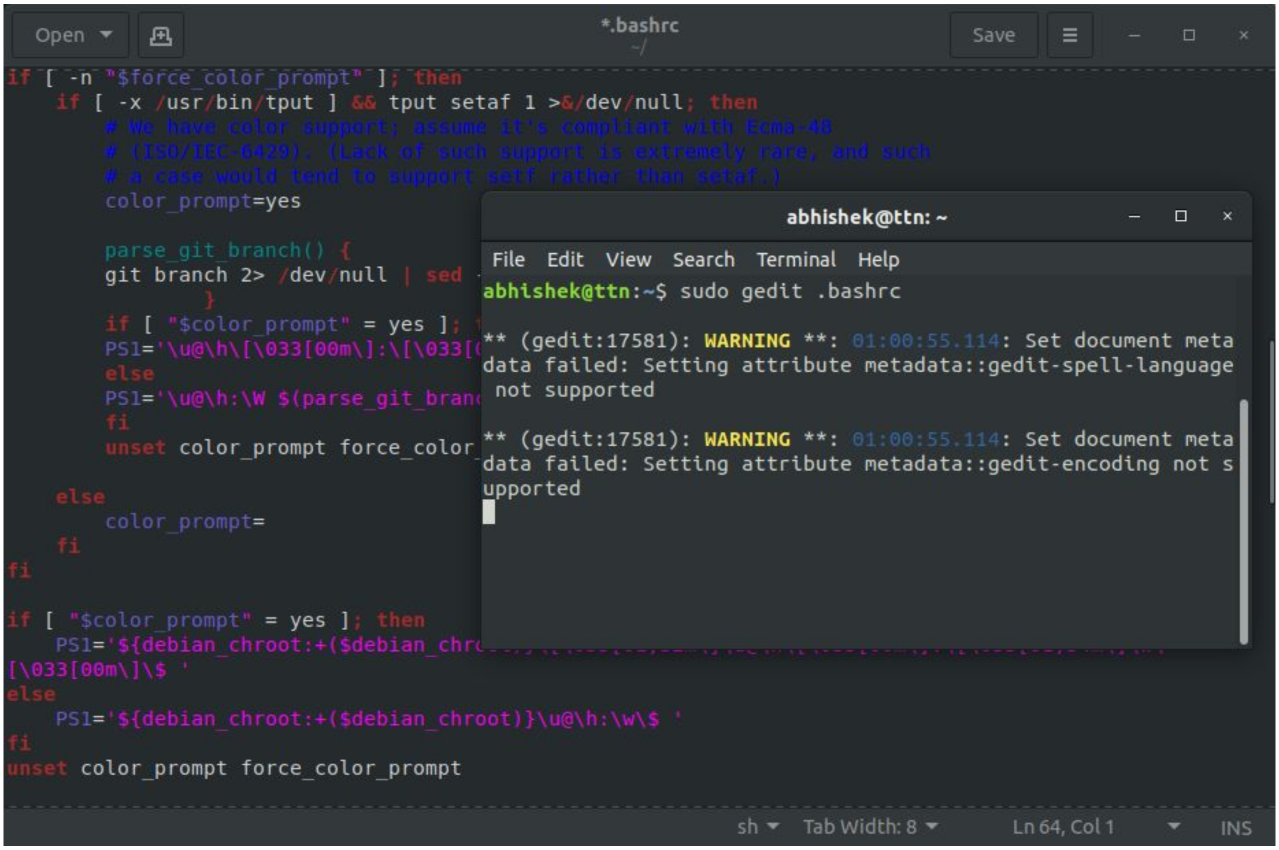


```
abhishek@ttn:~/Bootcamp/Exercise2$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
abhishek@ttn:~/Bootcamp/Exercise2$ echo "Hello world" > new
abhishek@ttn:~/Bootcamp/Exercise2$ git status
On branch master
Your branch is up to date with 'origin/master'.

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

        new

nothing added to commit but untracked files present (use "git add" to track)
abhishek@ttn:~/Bootcamp/Exercise2$ git add new
abhishek@ttn:~/Bootcamp/Exercise2$ git stash pop
No stash entries found.
```

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
```



```
if [ -n "$force_color_prompt" ]; then
    if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then
        # We have color support; assume it's compliant with Ecma-48
        # (ISO/IEC-6429). (Lack of such support is extremely rare, and such
        # a case would tend to support setf rather than setaf.)
        color_prompt=yes

        parse_git_branch() {
        git branch 2> /dev/null | sed
                }
        if [ "$color_prompt" = yes ];
        PS1='\u@\h\[\033[00m\]:\[\033[
        else
        PS1='\u@\h:\W $(parse_git_bran
        fi
        unset color_prompt force_color

    else
        color_prompt=
    fi
fi

if [ "$color_prompt" = yes ]; then
    PS1='${debian_chroot:+($debian_chr
[\033[00m\]\$ '
else
    PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
fi
unset color_prompt force_color_prompt
```

```
abhishek@ttn:~$ sudo gedit .bashrc

** (gedit:17581): WARNING **: 01:00:55.114: Set document meta
data failed: Setting attribute metadata::gedit-spell-language
 not supported

** (gedit:17581): WARNING **: 01:00:55.114: Set document meta
data failed: Setting attribute metadata::gedit-encoding not s
upported
```