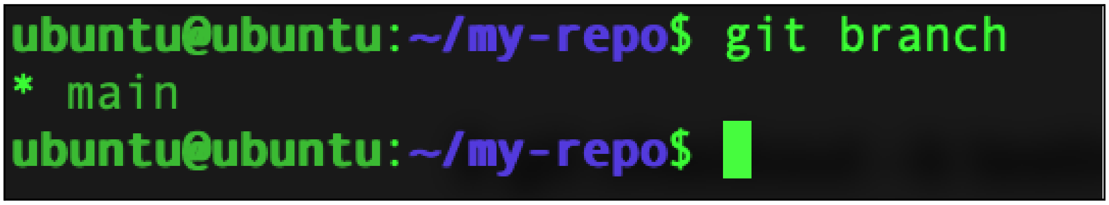
Objective : In this lab, you will create, merge and delete git branches.

Part 1: Branching

1. On your ubuntu remote workstation, in the **my-repo** directory, verify you are on the main branch:

   $ cd ~/my-repo

   $ git branch



```
ubuntu@ubuntu:~/my-repo$ git branch
* main
ubuntu@ubuntu:~/my-repo$
```

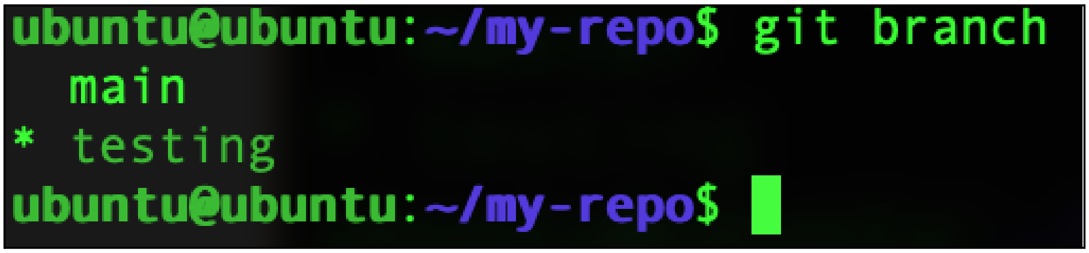
   Note: Current branch denoted by an **asterisk (*)**

2. Create and checkout to a new branch with the name **"testing"**:

   $ git checkout -b testing

3. Verify you are on the newly created branch:

   $ git branch



```
ubuntu@ubuntu:~/my-repo$ git branch
  main
* testing
ubuntu@ubuntu:~/my-repo$
```

4. Switch back to the main branch using following command :

   $ git checkout main

5. Verify you are on the main branch (noted by an asterisk):

```
$ git branch
```

6. Switch back to the **testing** branch

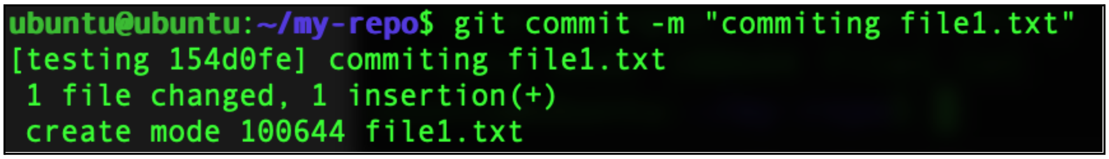
7. Verify you are on the testing branch

8. Create a file named **“file1.txt”** and add some content in it

9. Add **file1.txt** to the staging area of the testing branch with git add (refer back to the Lab **Basic Git Commands** if you don’t recall how to do this)

10. Commit the ‘**file1.txt**’ file

    Notice the branch name “testing” in the commit comments



```
ubuntu@ubuntu:~/my-repo$ git commit -m "commiting file1.txt"
[testing 154d0fe] commiting file1.txt
 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt
```

11. Look at the **commit logs** to see your commit message

    Note: Use ‘**q**’ to quit (if you need to exit from the 'git log' output)

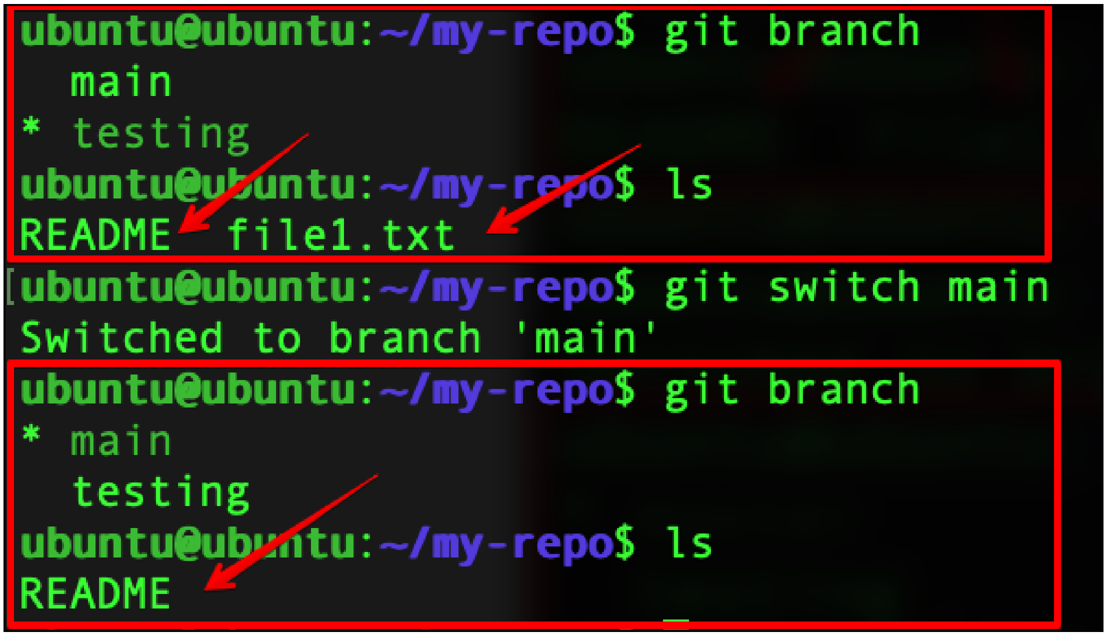
12. Look at the files in your directory. You will notice both files **README** and **file1.txt**:

```
$ ls -la
```

13. Switch back to the **main** branch

14. Look at the files in your directory. You should see only the **README file** Why?

15. Switch between **main** and **testing** branches a few times and watch the existence of **file1.txt** change, because it only **“belongs”** to the **testing** branch



```
ubuntu@ubuntu:~/my-repo$ git branch
  main
* testing
ubuntu@ubuntu:~/my-repo$ ls
README  file1.txt
[ubuntu@ubuntu:~/my-repo$ git switch main
Switched to branch 'main'
ubuntu@ubuntu:~/my-repo$ git branch
* main
  testing
ubuntu@ubuntu:~/my-repo$ ls
README
```

Part 2: Merging

16. Make sure you are on the **main** branch

17. Note the file contents of main branch only contains **README** and **.git:**

    $ ls -la

    The above command **'ls -la'** means:

    - list the contents of the current directory
    - show the long listing for each item (-l)
    - show all hidden files and directories (-a). Hidden files and directories in Linux start with a dot, such as '.git'

18. Merge the **testing** branch into the current **main** branch:

    $ git merge testing

    Note: Read the message showing the changes

19. To verify the files that the main branch contains, execute **ls -la** command. You will notice three files **file1.txt, README and .git**

20. Compare contents of both branches, note they are now identical. Using **'git checkout'** might be useful for this

Part 3: Deleting

21. Create a new branch with the name "**newbranch**". For demonstration, this will be deleted in upcoming steps

22. Verify that you are on the newly created branch

    The list of branches includes:
    main, newbranch and testing

23. To delete the **'newbranch'** branch:

    ```
    $ git branch -D newbranch
    ```

    Note: You can't delete the current branch. You have to first checkout a different branch in order to delete the current branch

24. Change to the **testing** branch, then delete the **newbranch** branch

25. Verify your current branch and check whether the **"newbranch"** is deleted from the branch list

Notify your instructor that you are done with the Lab

END OF LAB