Lesson 01 Demo 05

Switching Branches in Git

Objective: To switch between branches in Git and manage different features, fixes, or versions of a project without merging changes immediately

Prerequisite: GitHub Account

Tools required: Git

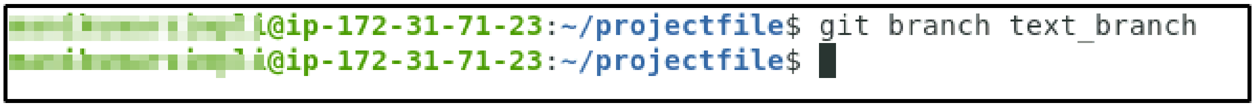
Steps to be followed:

1. Create a new branch
2. Switch to the new branch
3. Create a file and commit the changes
4. Check the status of the new branch
5. Switch back to the main branch

> **Note:** A **projectfile** is already created in Demo 04. If you do not have a repository, first perform that demo and then execute the steps from this demo.

Step 1: Create a new branch

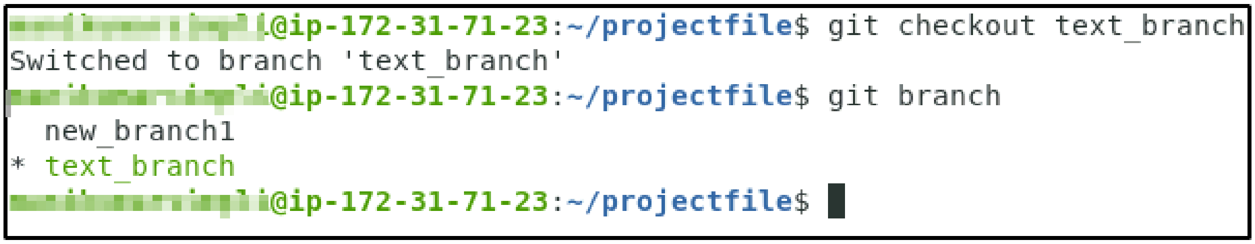
1.1 Open the terminal and execute the following command to create a new branch:
git branch text_branch



```
[redacted]i@ip-172-31-71-23:~/projectfile$ git branch text_branch
[redacted]i@ip-172-31-71-23:~/projectfile$ 
```

Step 2: Switch to the new branch

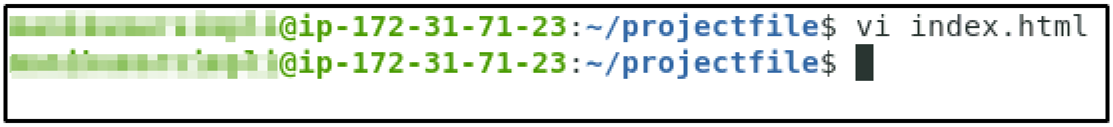
2.1 Use the following command to switch to the newly created text_branch:
git checkout text_branch



```
[redacted]@ip-172-31-71-23:~/projectfile$ git checkout text_branch
Switched to branch 'text_branch'
[redacted]@ip-172-31-71-23:~/projectfile$ git branch
  new_branch1
* text_branch
[redacted]@ip-172-31-71-23:~/projectfile$ 
```

Step 3: Create a file and commit the changes

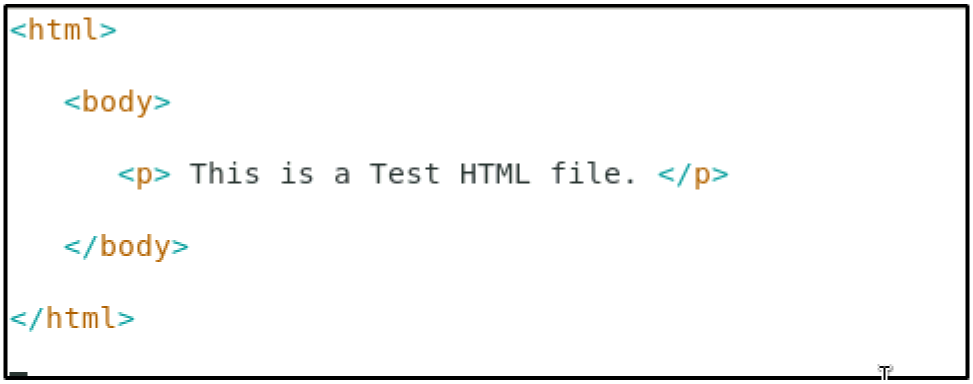
3.1 Use the following command to create a file:
vi index.html



```
[redacted]@ip-172-31-71-23:~/projectfile$ vi index.html
[redacted]@ip-172-31-71-23:~/projectfile$ 
```

3.2 Add the following code snippet in the **index.html** file:

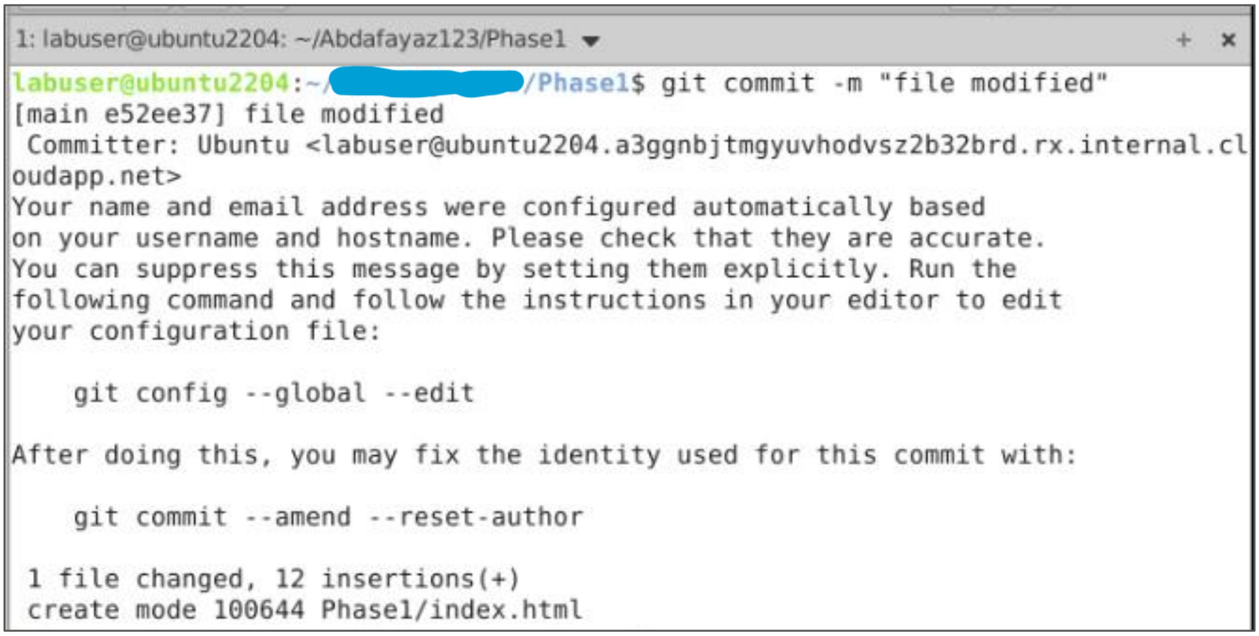
```
<html>
  <body>
     <p> This is a Test HTML file. </p>
  </body>
</html>
```



```
<html>

   <body>

      <p> This is a Test HTML file. </p>

   </body>

</html>
```

Note: Press **Esc** button and enter **:wq** to save the file

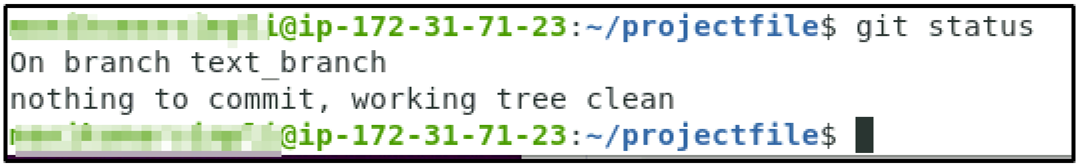
3.3 Use the following command to add the file to the text_branch**,** and commit the changes:
git commit -m "file modified"



```
labuser@ubuntu2204:~/[redacted]/Phase1$ git commit -m "file modified"
[main e52ee37] file modified
 Committer: Ubuntu <labuser@ubuntu2204.a3ggnbjtmgyuvhodvsz2b32brd.rx.internal.cl
oudapp.net>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 12 insertions(+)
 create mode 100644 Phase1/index.html
```

Step 4: Check the status of the new branch

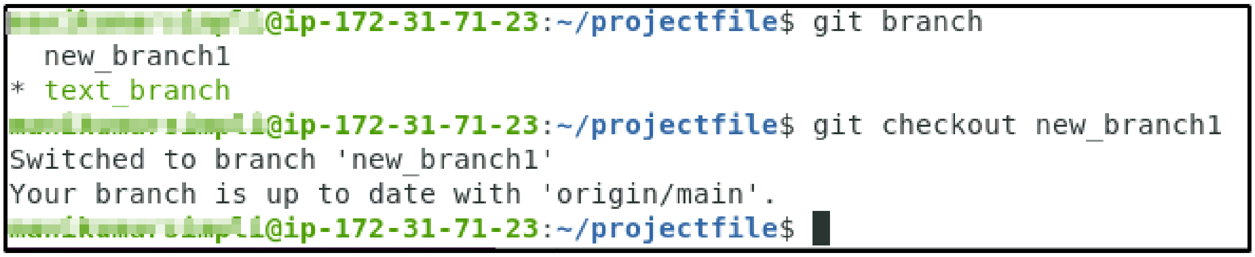
4.1 Check the status of the new branch using the following command:
git status



```
[redacted]@ip-172-31-71-23:~/projectfile$ git status
On branch text_branch
nothing to commit, working tree clean
[redacted]@ip-172-31-71-23:~/projectfile$ 
```

Step 5: Switch back to the main branch

5.1 Check the current branch and switch back to the main branch using the following commands:
git branch
git checkout new_branch1



```
[redacted]@ip-172-31-71-23:~/projectfile$ git branch
  new_branch1
* text_branch
[redacted]@ip-172-31-71-23:~/projectfile$ git checkout new_branch1
Switched to branch 'new_branch1'
Your branch is up to date with 'origin/main'.
[redacted]@ip-172-31-71-23:~/projectfile$ 
```

By following these steps, you have successfully switched between branches in Git to manage different features, fixes, or versions of a project without merging changes immediately.