

Lesson 04 Demo 01

Working with Branches

Objective: To understand how to create, manage, rename, and delete branches in GitHub using Git commands, enhancing your ability to organize and collaborate on code efficiently

Tools Required: Git

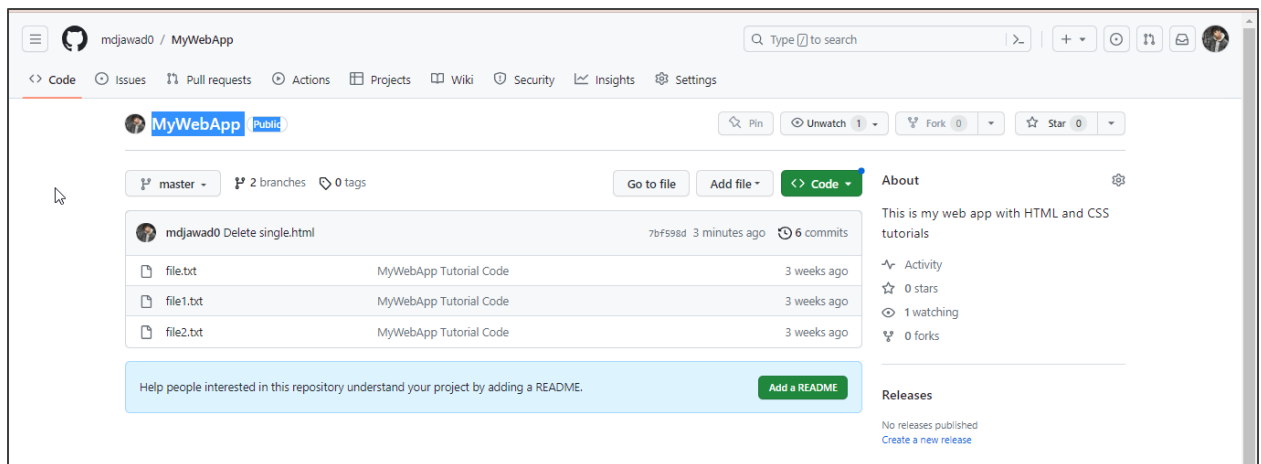
Prerequisites: Lesson 05 Demo 01

Steps to be followed:

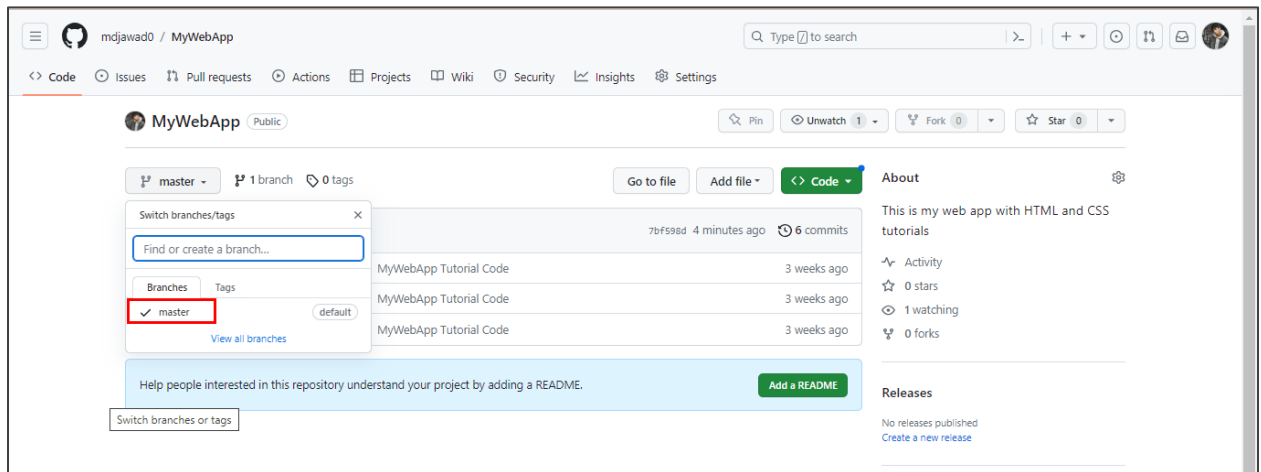
1. Create branches using GitHub
2. Work with branches using Git commands
3. Rename and delete branches

Step 1: Create branches using GitHub

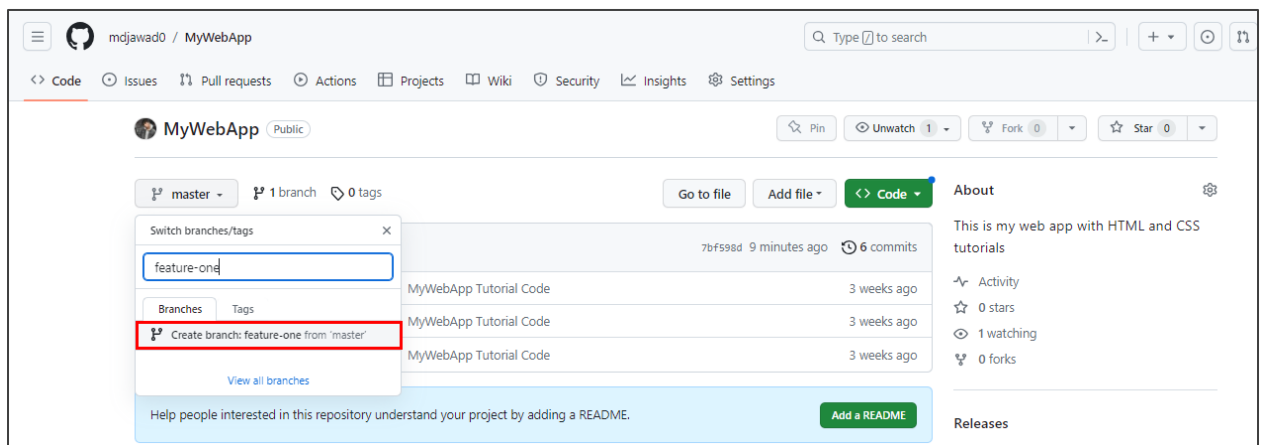
1.1 Open the GitHub repository and locate your project



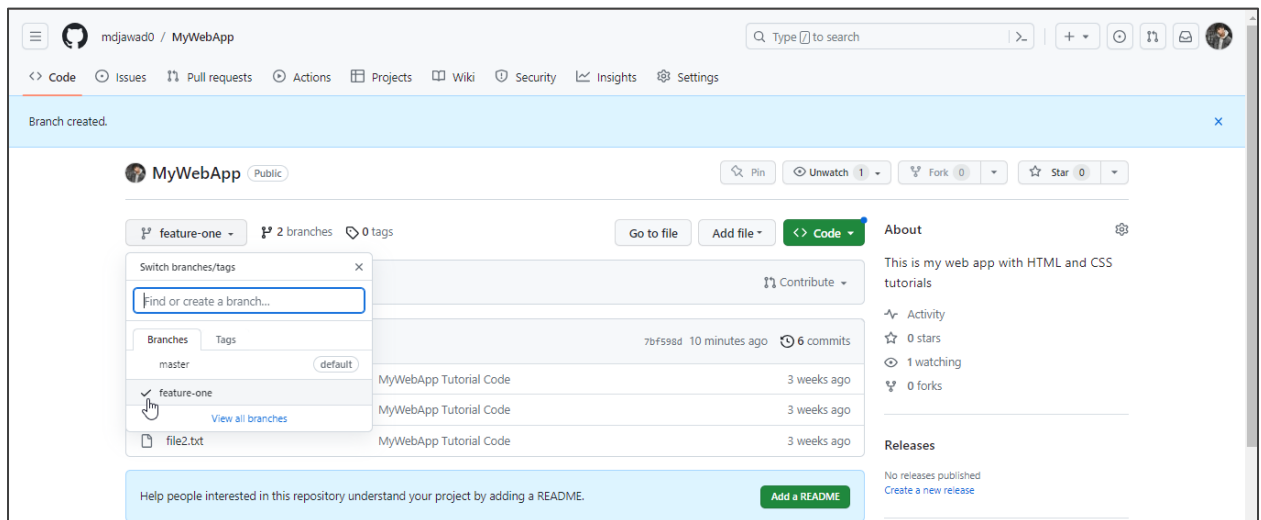
1.2 Verify that a branch called **master** already exists in the repository



1.3 Create a new branch using the GitHub interface. Name the new branch as **feature-one** and click **Create**.

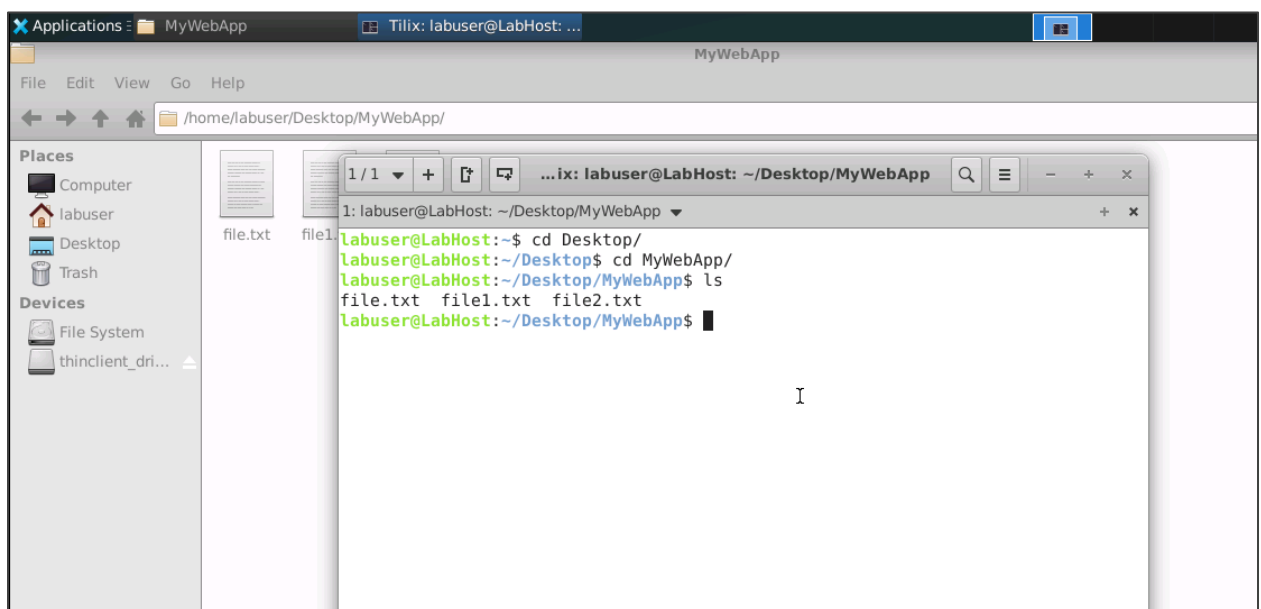


1.4 Verify the creation of the new branch

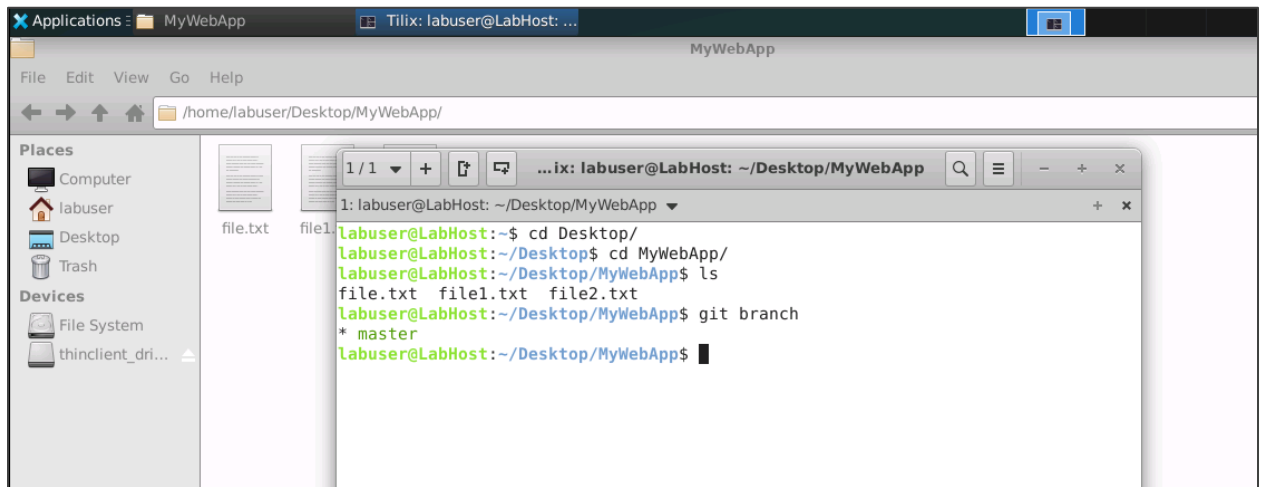


Step 2: Work with branches using Git commands

2.1 Open the terminal and navigate to the local Git repository

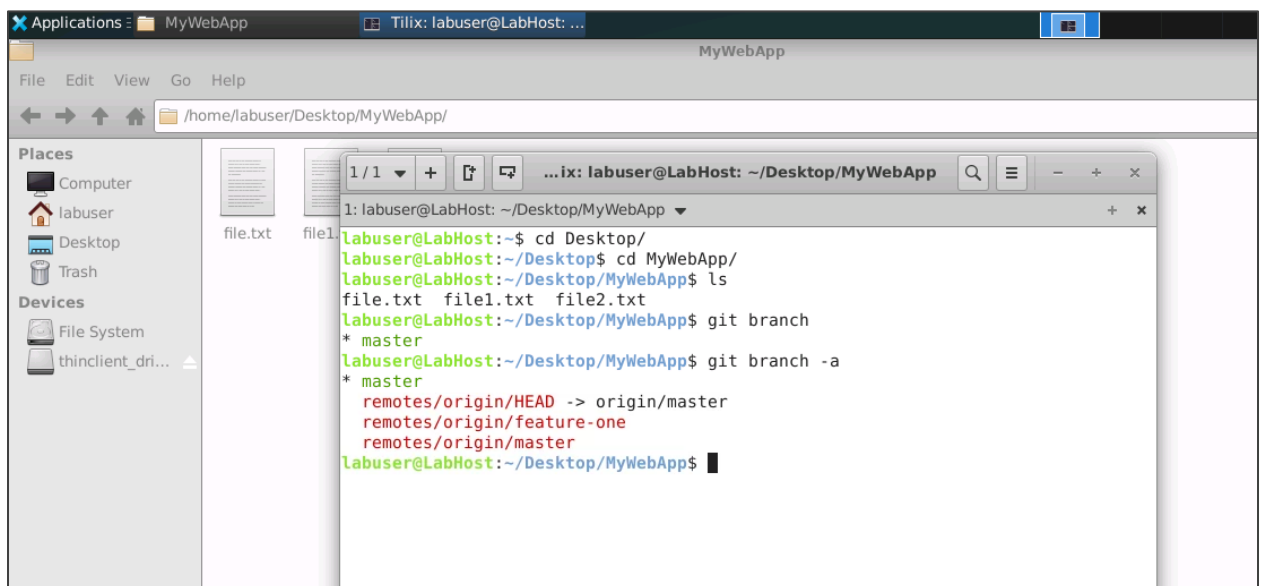


2.2 List all local branches using **git branch**



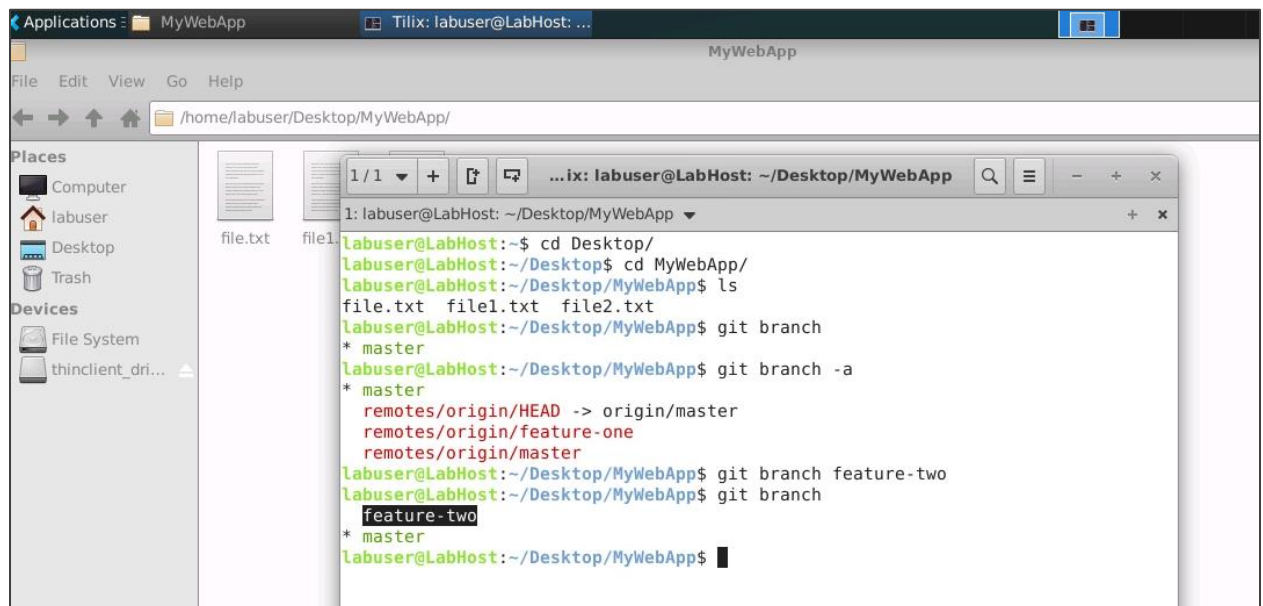
```
labuser@LabHost: ~/$ cd Desktop/  
labuser@LabHost:~/Desktop$ cd MyWebApp/  
labuser@LabHost:~/Desktop/MyWebApp$ ls  
file.txt file1.txt file2.txt  
labuser@LabHost:~/Desktop/MyWebApp$ git branch  
* master  
labuser@LabHost:~/Desktop/MyWebApp$
```

2.3 List all local and remote branches using **git branch -a**



```
labuser@LabHost: ~/$ cd Desktop/  
labuser@LabHost:~/Desktop$ cd MyWebApp/  
labuser@LabHost:~/Desktop/MyWebApp$ ls  
file.txt file1.txt file2.txt  
labuser@LabHost:~/Desktop/MyWebApp$ git branch  
* master  
labuser@LabHost:~/Desktop/MyWebApp$ git branch -a  
* master  
remotes/origin/HEAD -> origin/master  
remotes/origin/feature-one  
remotes/origin/master  
labuser@LabHost:~/Desktop/MyWebApp$
```

2.4 Create a new local branch using `git branch <branch-name>`

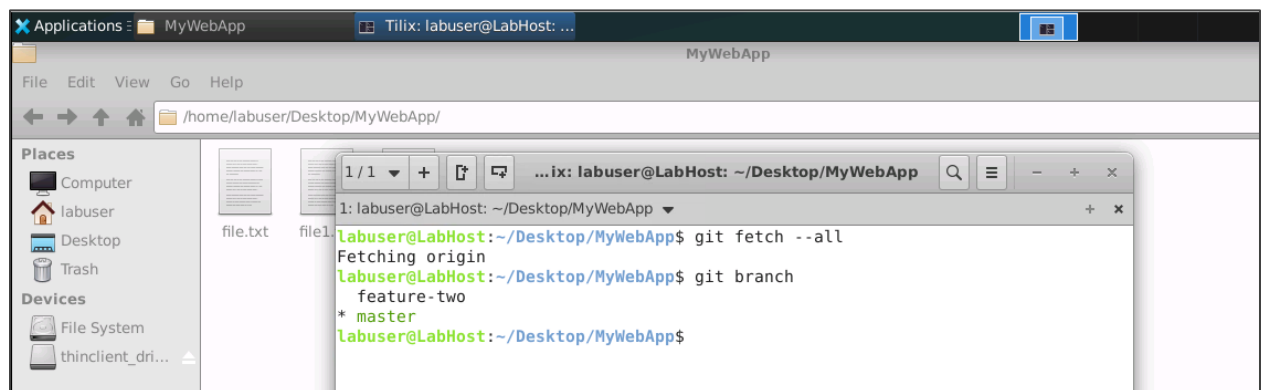


```
Applications ▾ MyWebApp Tilix: labuser@LabHost: ... MyWebApp
File Edit View Go Help
/home/labuser/Desktop/MyWebApp/

Places
Computer
labuser
Desktop
Trash
Devices
File System
thinclient_dri...

file.txt file1
1 / 1 + ...ix: labuser@LabHost: ~/Desktop/MyWebApp
1: labuser@LabHost: ~/Desktop/MyWebApp ▾
labuser@LabHost:~$ cd Desktop/
labuser@LabHost:~/Desktop$ cd MyWebApp/
labuser@LabHost:~/Desktop/MyWebApp$ ls
file.txt file1.txt file2.txt
labuser@LabHost:~/Desktop/MyWebApp$ git branch
* master
labuser@LabHost:~/Desktop/MyWebApp$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/feature-one
  remotes/origin/master
labuser@LabHost:~/Desktop/MyWebApp$ git branch feature-two
labuser@LabHost:~/Desktop/MyWebApp$ git branch
feature-two
* master
labuser@LabHost:~/Desktop/MyWebApp$
```

2.5 Sync local and remote branches using `git fetch --all`

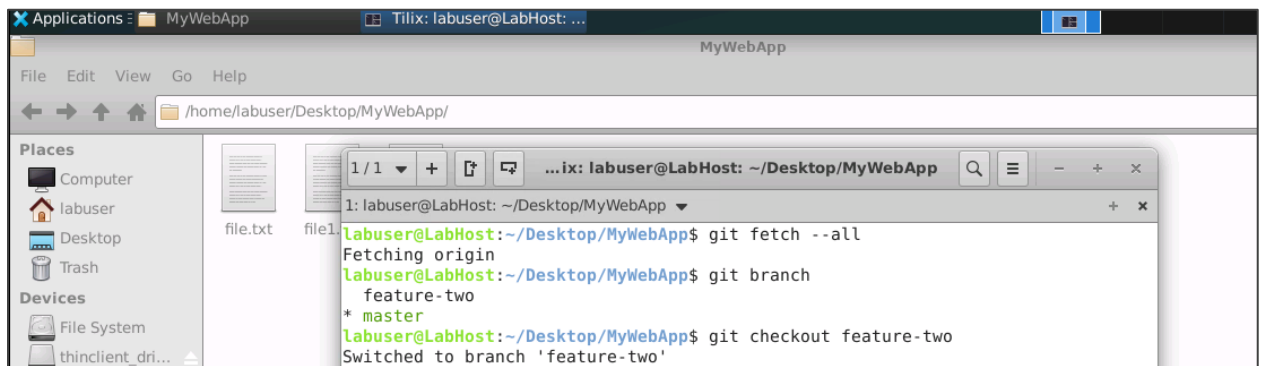


```
Applications ▾ MyWebApp Tilix: labuser@LabHost: ... MyWebApp
File Edit View Go Help
/home/labuser/Desktop/MyWebApp/

Places
Computer
labuser
Desktop
Trash
Devices
File System
thinclient_dri...

file.txt file1
1 / 1 + ...ix: labuser@LabHost: ~/Desktop/MyWebApp
1: labuser@LabHost: ~/Desktop/MyWebApp ▾
labuser@LabHost:~/Desktop/MyWebApp$ git fetch --all
Fetching origin
labuser@LabHost:~/Desktop/MyWebApp$ git branch
feature-two
* master
labuser@LabHost:~/Desktop/MyWebApp$
```

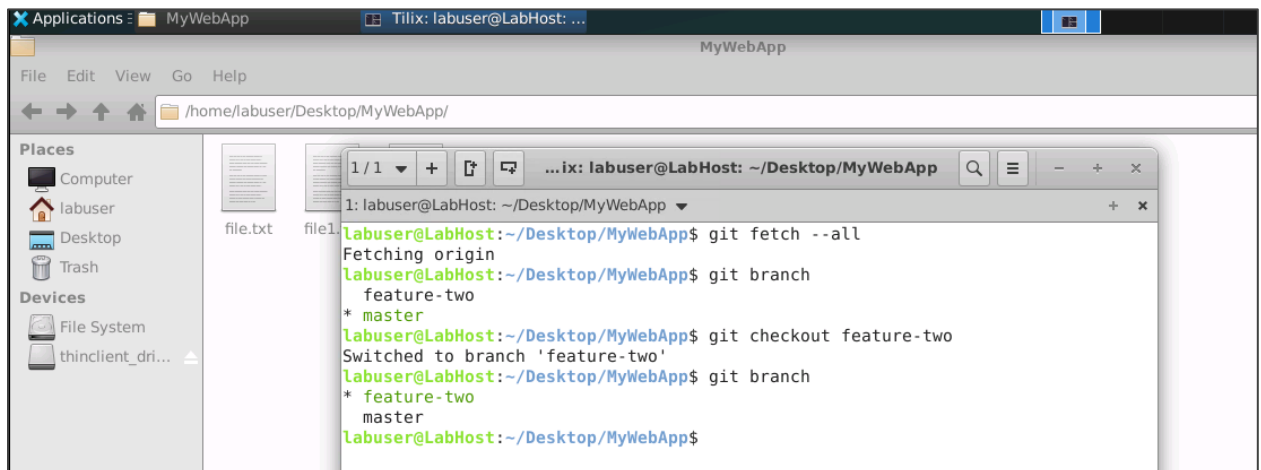
2.6 Switch to a specific branch using `git checkout <branch-name>`



The screenshot shows a terminal window titled 'Tilix: labuser@LabHost: ...' with the working directory set to '/home/labuser/Desktop/MyWebApp/'. The terminal displays the following commands and output:

```
labuser@LabHost: ~/Desktop/MyWebApp$ git fetch --all
Fetching origin
labuser@LabHost: ~/Desktop/MyWebApp$ git branch
* master
labuser@LabHost: ~/Desktop/MyWebApp$ git checkout feature-two
Switched to branch 'feature-two'
```

2.7 Verify the branch switch

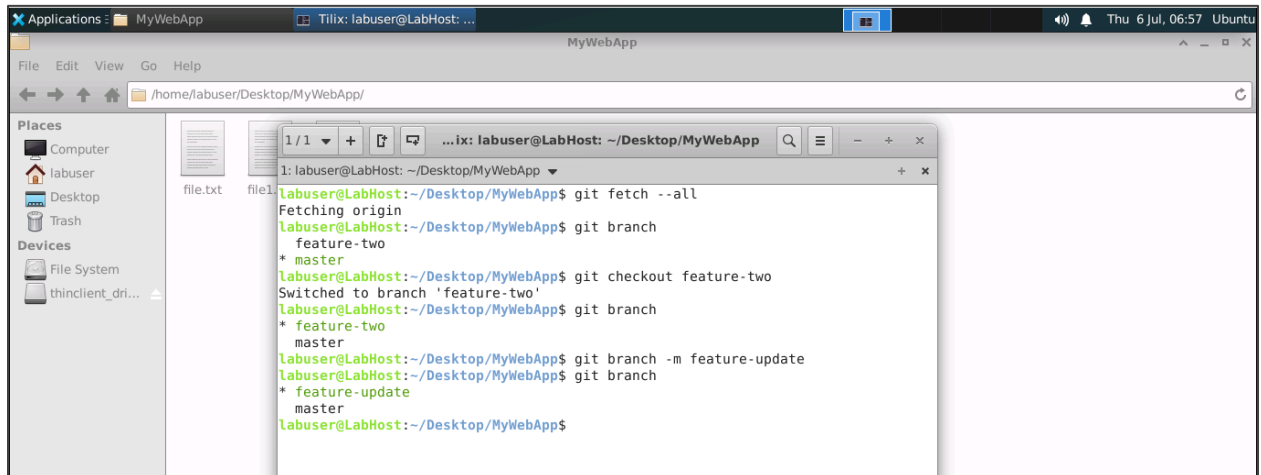


The screenshot shows the same terminal window as in the previous step, but with an additional command to verify the branch switch:

```
labuser@LabHost: ~/Desktop/MyWebApp$ git branch
* feature-two
  master
labuser@LabHost: ~/Desktop/MyWebApp$
```

Step 3: Rename and delete branches

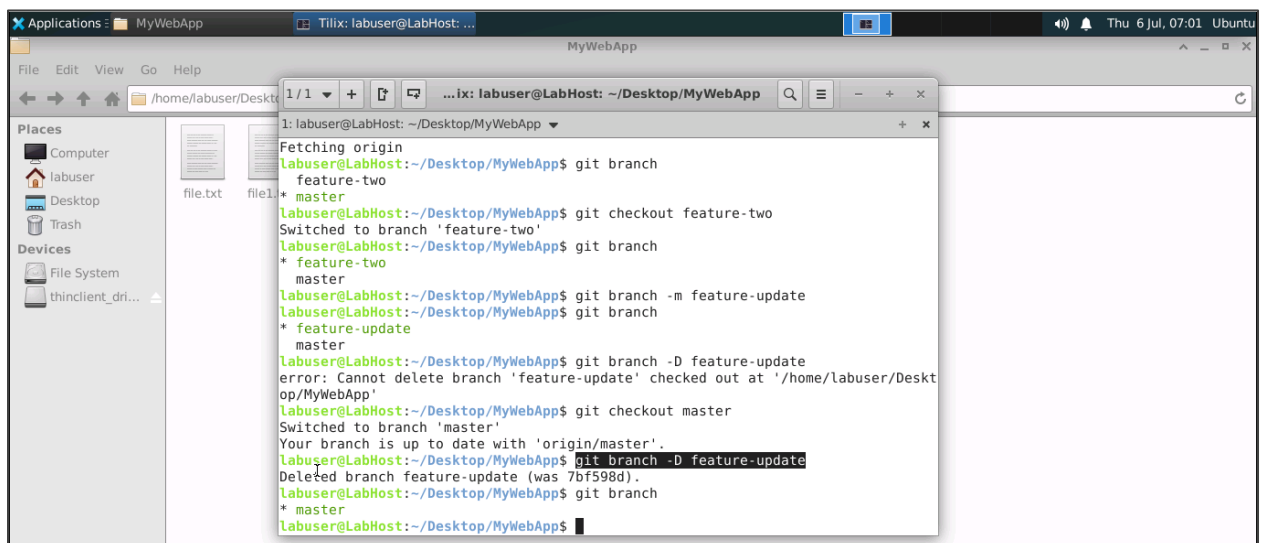
3.1 Rename a branch using **git branch -m <new-name>**. Verify the branch rename.



The screenshot shows a terminal window titled 'Tilix: labuser@LabHost: ...' with the working directory set to `~/Desktop/MyWebApp`. The terminal displays the following commands and their output:

```
labuser@LabHost:~/Desktop/MyWebApp$ git fetch --all
Fetching origin
labuser@LabHost:~/Desktop/MyWebApp$ git branch
* master
labuser@LabHost:~/Desktop/MyWebApp$ git checkout feature-two
Switched to branch 'feature-two'
labuser@LabHost:~/Desktop/MyWebApp$ git branch
* feature-two
  master
labuser@LabHost:~/Desktop/MyWebApp$ git branch -m feature-update
labuser@LabHost:~/Desktop/MyWebApp$ git branch
* feature-update
  master
labuser@LabHost:~/Desktop/MyWebApp$
```

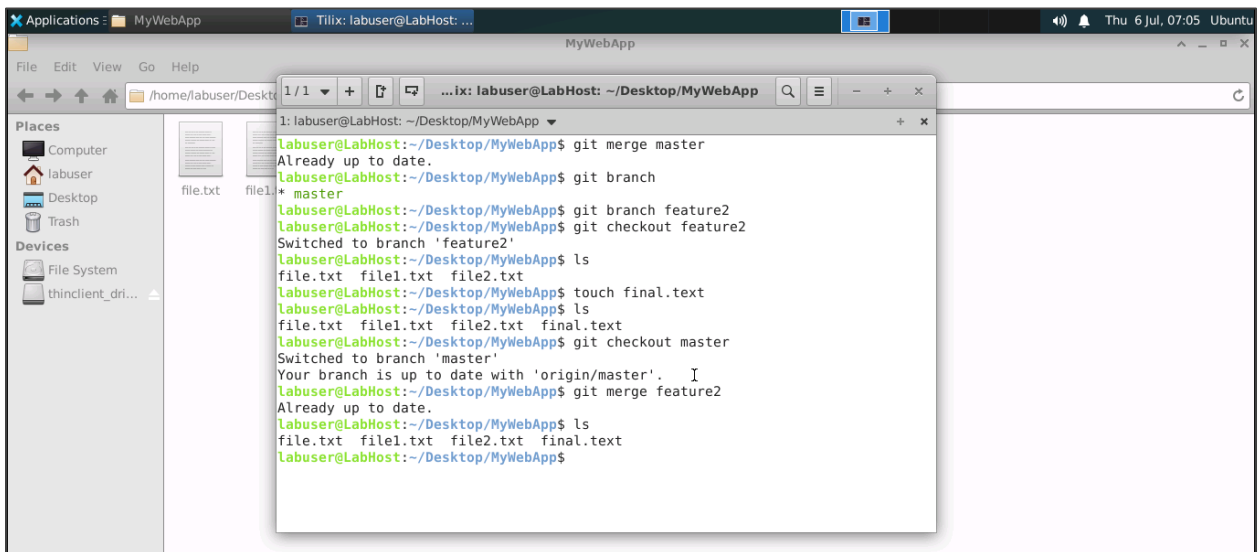
3.2 Delete a branch using **git branch -D <branch-name>**. Verify the branch deletion.



The screenshot shows the same terminal window as before, but with additional commands to delete a branch and return to master:

```
labuser@LabHost:~/Desktop/MyWebApp$ git branch
* feature-two
  master
labuser@LabHost:~/Desktop/MyWebApp$ git checkout feature-two
Switched to branch 'feature-two'
labuser@LabHost:~/Desktop/MyWebApp$ git branch
* feature-two
  master
labuser@LabHost:~/Desktop/MyWebApp$ git branch -m feature-update
labuser@LabHost:~/Desktop/MyWebApp$ git branch
* feature-update
  master
labuser@LabHost:~/Desktop/MyWebApp$ git branch -D feature-update
error: Cannot delete branch 'feature-update' checked out at '/home/labuser/Desktop/MyWebApp'
labuser@LabHost:~/Desktop/MyWebApp$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
labuser@LabHost:~/Desktop/MyWebApp$ git branch -D feature-update
Deleted branch feature-update (was 7bf598d).
labuser@LabHost:~/Desktop/MyWebApp$ git branch
* master
labuser@LabHost:~/Desktop/MyWebApp$
```

3.3 To merge changes from a source branch, use `git merge <source-branch>`



```
1: labuser@LabHost: ~/Desktop/MyWebApp
labuser@LabHost:~/Desktop/MyWebApp$ git merge master
Already up to date.
labuser@LabHost:~/Desktop/MyWebApp$ git branch
* master
labuser@LabHost:~/Desktop/MyWebApp$ git branch feature2
labuser@LabHost:~/Desktop/MyWebApp$ git checkout feature2
Switched to branch 'feature2'
labuser@LabHost:~/Desktop/MyWebApp$ ls
file.txt  file1.txt  file2.txt
labuser@LabHost:~/Desktop/MyWebApp$ touch final.txt
labuser@LabHost:~/Desktop/MyWebApp$ ls
file.txt  file1.txt  file2.txt  final.txt
labuser@LabHost:~/Desktop/MyWebApp$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
labuser@LabHost:~/Desktop/MyWebApp$ git merge feature2
Already up to date.
labuser@LabHost:~/Desktop/MyWebApp$ ls
file.txt  file1.txt  file2.txt  final.txt
labuser@LabHost:~/Desktop/MyWebApp$
```

By following these steps, you have learned how to effectively create, manage, rename, and delete branches in GitHub using Git commands, which will help you maintain a well-organized and efficient workflow in your projects.