Module 2 - Git Assignment 2

Tasks To Be Performed:

1. Create a Git working directory with feature1.txt and feature2.txt in the master branch

2. Create 3 branches develop, feature1 and feature2

3. In develop branch create develop.txt, do not stage or commit it

4. Stash this file and check out to feature1 branch

5. Create new.txt file in feature1 branch, stage and commit this file

6. Checkout to develop, unstash this file and commit

7. Please submit all the Git commands used to do the above steps

Steps

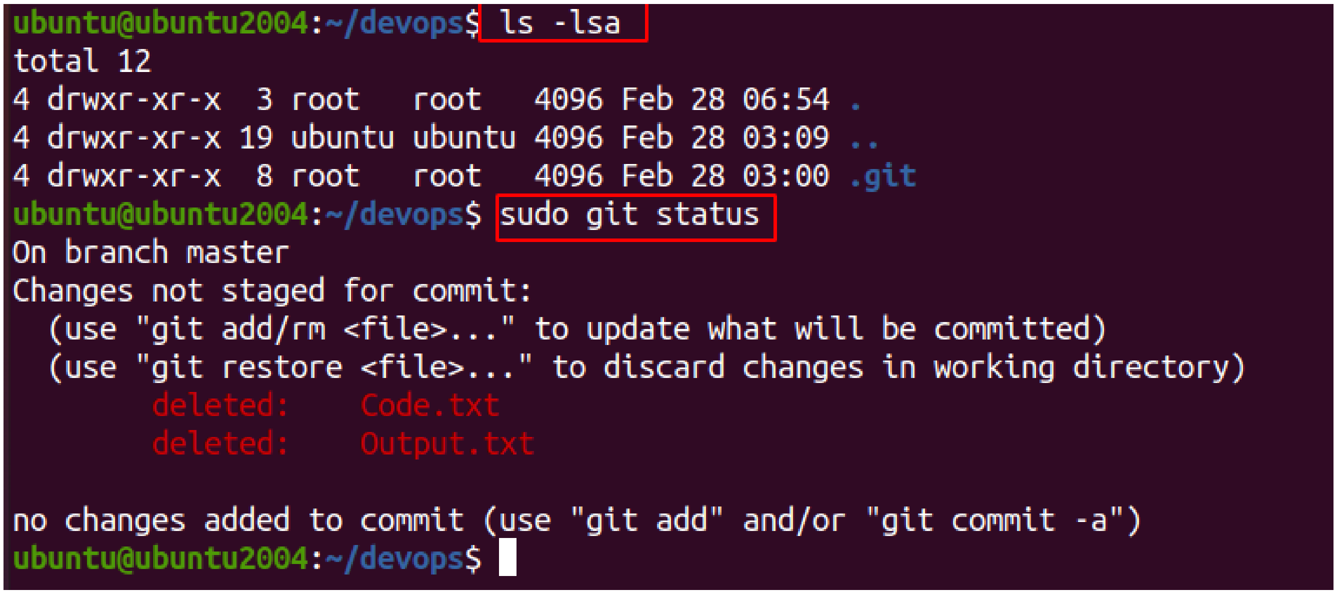
1. Navigate to the Ubuntu terminal

2. Create a new directory git2 using the command sudo mkdir git2 or use the same directory created during assignment 1

3. Initialize the folder using git init if a new directory is created

4. I will use the same folder what I created earlier in assignment 1

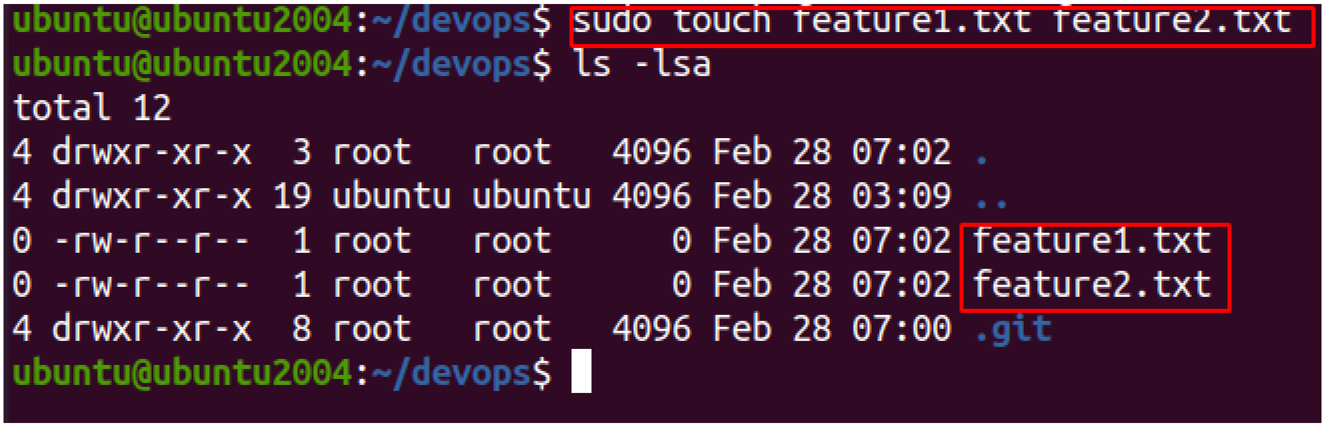
5. The folder devops is already initialized and I will not re-run the command.



```
ubuntu@ubuntu2004:~/devops$ ls -lsa
total 12
4 drwxr-xr-x  3 root   root   4096 Feb 28 06:54 .
4 drwxr-xr-x 19 ubuntu ubuntu 4096 Feb 28 03:09 ..
4 drwxr-xr-x  8 root   root   4096 Feb 28 03:00 .git
ubuntu@ubuntu2004:~/devops$ sudo git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    Code.txt
        deleted:    Output.txt

no changes added to commit (use "git add" and/or "git commit -a")
ubuntu@ubuntu2004:~/devops$ 
```

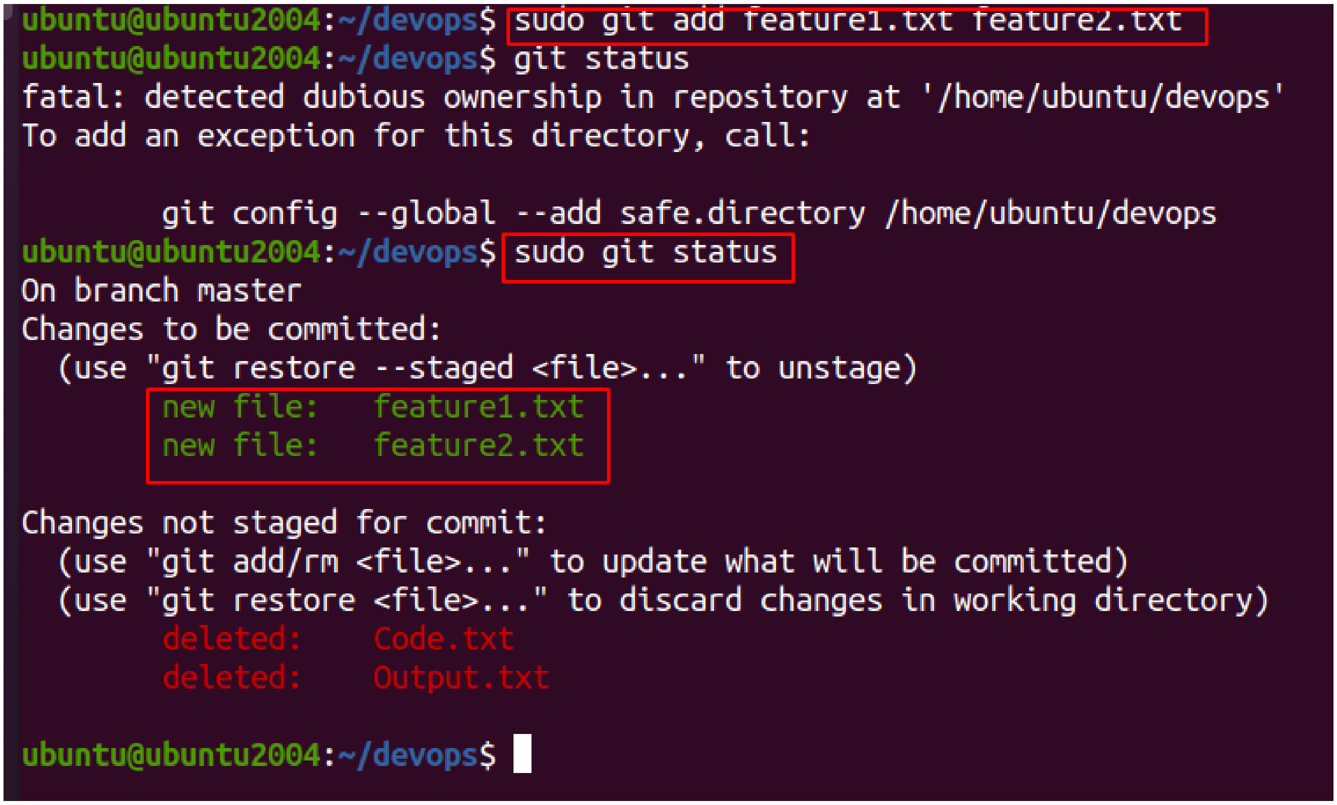
6. I will create two text files feature 1 and feature 2 using the terminal

7. sudo touch feature1.txt feature2.txt



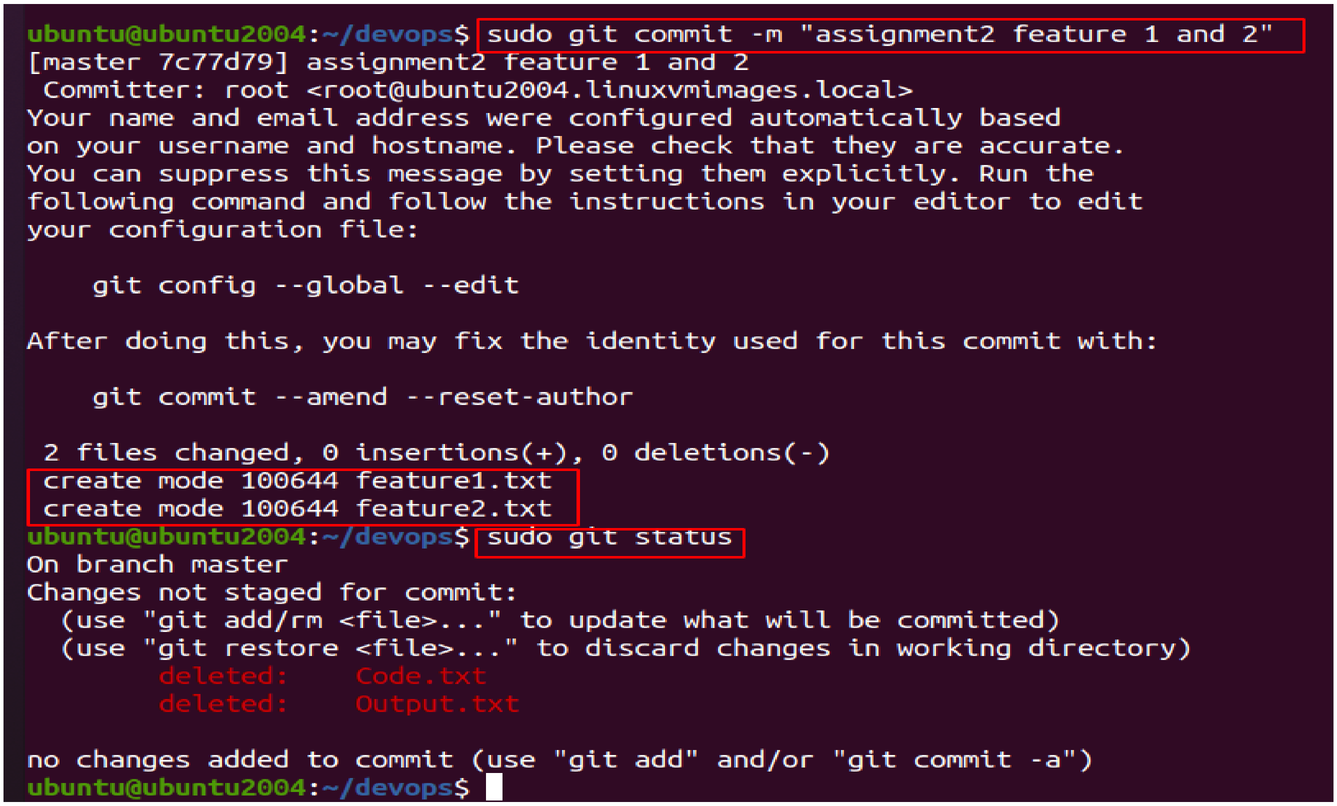
```
ubuntu@ubuntu2004:~/devops$ sudo touch feature1.txt feature2.txt
ubuntu@ubuntu2004:~/devops$ ls -lsa
total 12
4 drwxr-xr-x  3 root   root   4096 Feb 28 07:02 .
4 drwxr-xr-x 19 ubuntu ubuntu 4096 Feb 28 03:09 ..
0 -rw-r--r--  1 root   root      0 Feb 28 07:02 feature1.txt
0 -rw-r--r--  1 root   root      0 Feb 28 07:02 feature2.txt
4 drwxr-xr-x  8 root   root   4096 Feb 28 07:00 .git
ubuntu@ubuntu2004:~/devops$ 
```

8. Now I will run the command **git add .** The . Will include all the files in the folder to be committed or we can do individually as in assignment 1.



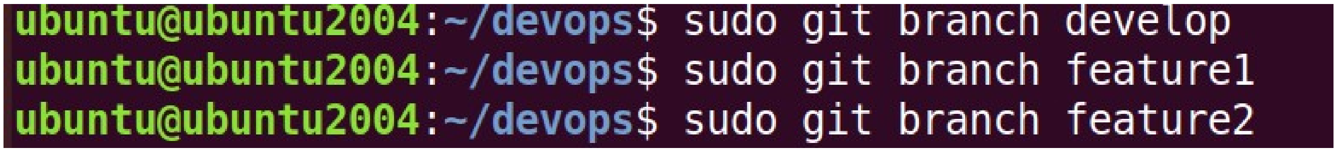
```
ubuntu@ubuntu2004:~/devops$ sudo git add feature1.txt feature2.txt
ubuntu@ubuntu2004:~/devops$ git status
fatal: detected dubious ownership in repository at '/home/ubuntu/devops'
To add an exception for this directory, call:

        git config --global --add safe.directory /home/ubuntu/devops
ubuntu@ubuntu2004:~/devops$ sudo git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   feature1.txt
        new file:   feature2.txt

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    Code.txt
        deleted:    Output.txt

ubuntu@ubuntu2004:~/devops$ 
```

9. Now I will commit the files – feature1.txt and feature2.txt



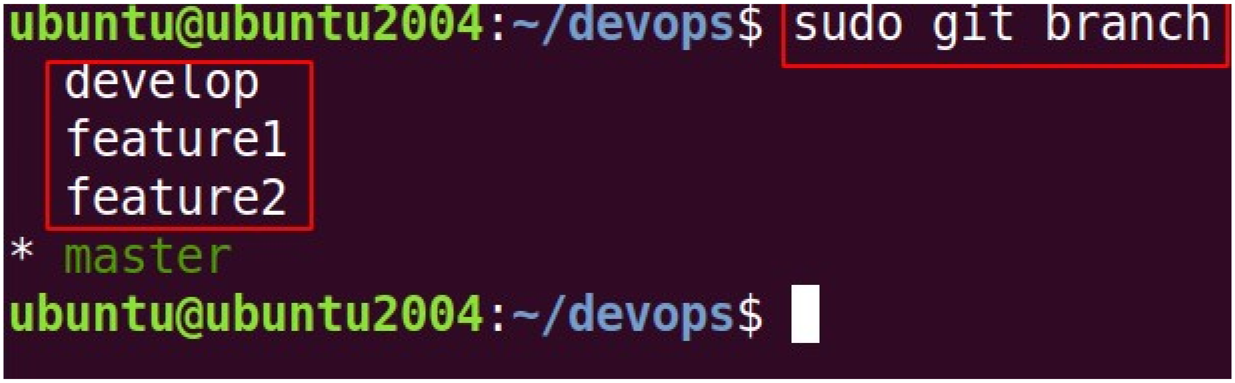
```
ubuntu@ubuntu2004:~/devops$ sudo git commit -m "assignment2 feature 1 and 2"
[master 7c77d79] assignment2 feature 1 and 2
 Committer: root <root@ubuntu2004.linuxvmimages.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are 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

 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 feature1.txt
 create mode 100644 feature2.txt
ubuntu@ubuntu2004:~/devops$ sudo git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    Code.txt
        deleted:    Output.txt

no changes added to commit (use "git add" and/or "git commit -a")
ubuntu@ubuntu2004:~/devops$ 
```

10. Now we need to create a branch
1. branch – develop
2. branch – feature1
3. branch – feature2



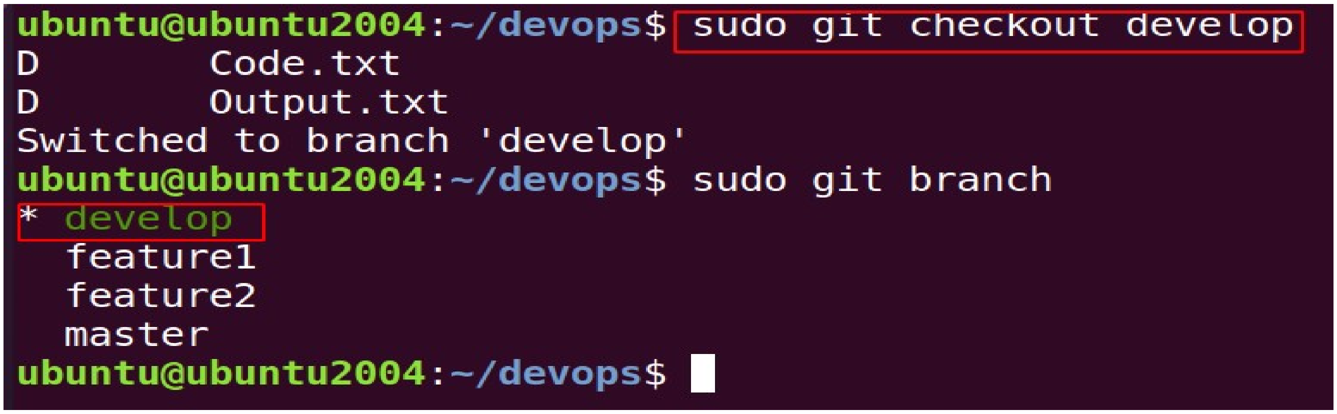
```
ubuntu@ubuntu2004:~/devops$ sudo git branch develop
ubuntu@ubuntu2004:~/devops$ sudo git branch feature1
ubuntu@ubuntu2004:~/devops$ sudo git branch feature2
```

11. The branches are created and run the command sudo git branch and you can see the branches created.



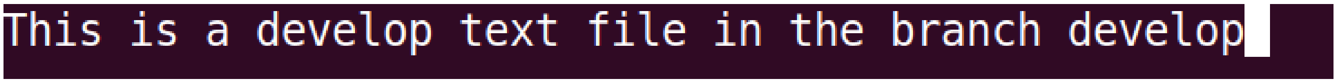
```
ubuntu@ubuntu2004:~/devops$ sudo git branch
  develop
  feature1
  feature2
* master
ubuntu@ubuntu2004:~/devops$ 
```

12. Notice the asterix and we are currently in the master branch. We need to chekout to the develop branch and create a text file called develop.txt. No need to stage or commit it.

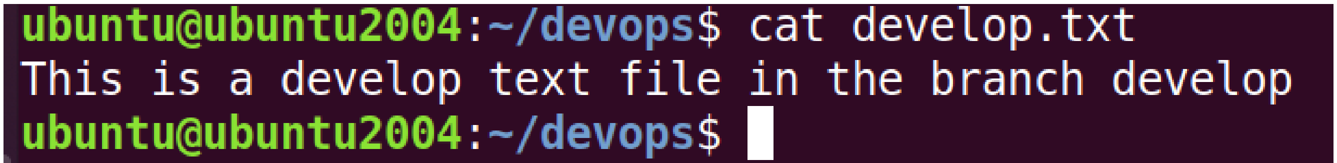


```
ubuntu@ubuntu2004:~/devops$ sudo git checkout develop
D       Code.txt
D       Output.txt
Switched to branch 'develop'
ubuntu@ubuntu2004:~/devops$ sudo git branch
* develop
  feature1
  feature2
  master
ubuntu@ubuntu2004:~/devops$ 
```

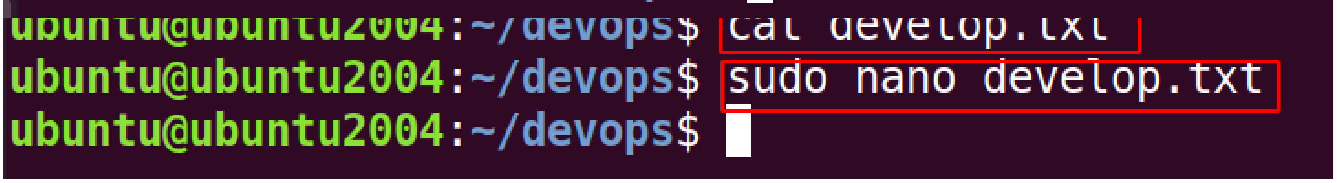
13. The next requirement is not to stage or commit but to stash the file. So we will create a text file in the develop branch and stash it . The command to stash is git stash



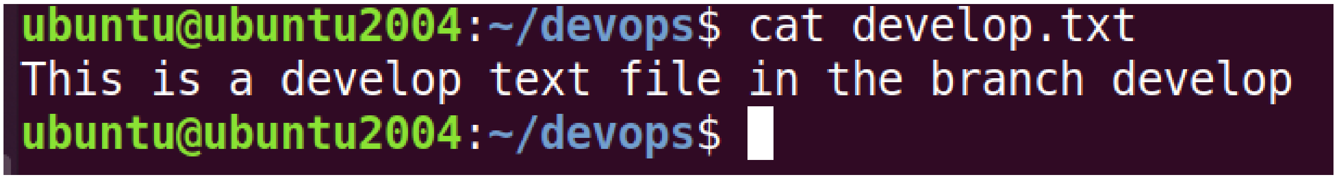
```
This is a develop text file in the branch develop
```



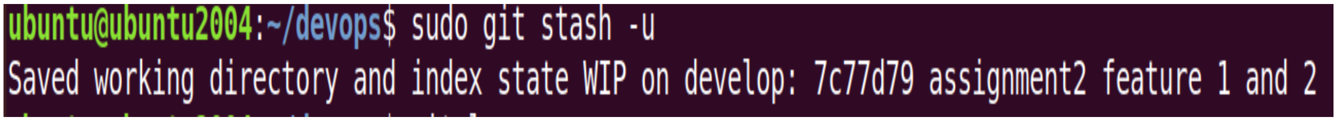
```
ubuntu@ubuntu2004:~/devops$ cat develop.txt
This is a develop text file in the branch develop
ubuntu@ubuntu2004:~/devops$
```



```
ubuntu@ubuntu2004:~/devops$ cat develop.txt
ubuntu@ubuntu2004:~/devops$ sudo nano develop.txt
ubuntu@ubuntu2004:~/devops$
```

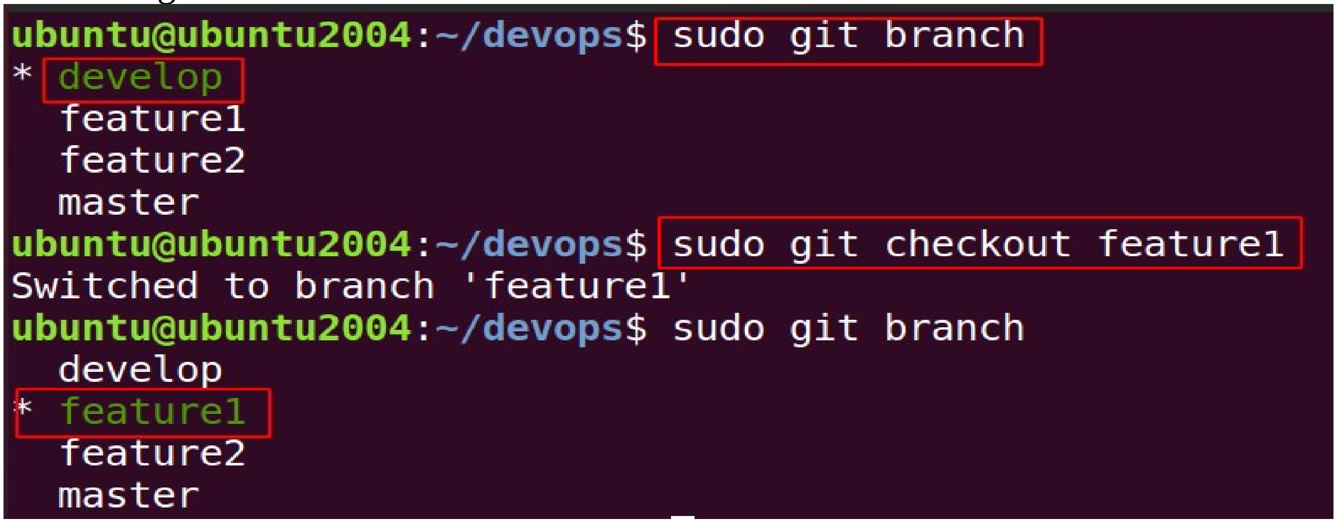


```
ubuntu@ubuntu2004:~/devops$ cat develop.txt
This is a develop text file in the branch develop
ubuntu@ubuntu2004:~/devops$
```



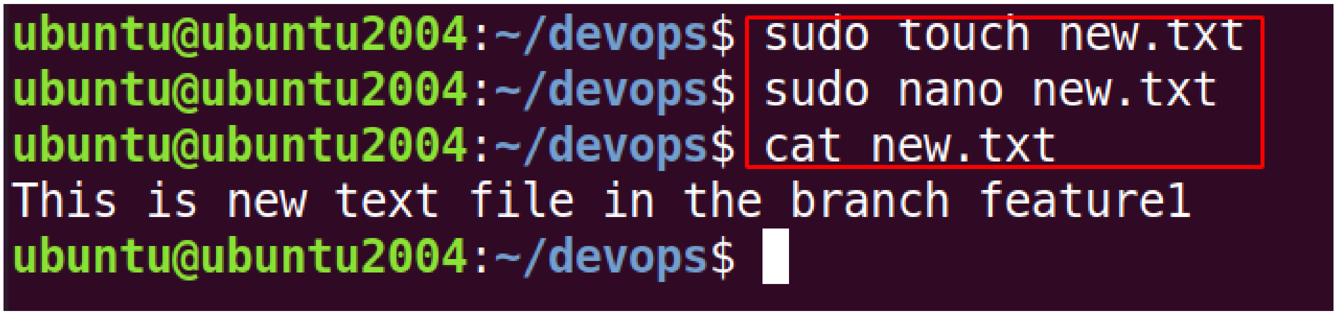
```
ubuntu@ubuntu2004:~/devops$ sudo git stash -u
Saved working directory and index state WIP on develop: 7c77d79 assignment2 feature 1 and 2
```

14, Now checkout to the branch feature1 and create a new.txt file and the new.txt file will be staged and committed.



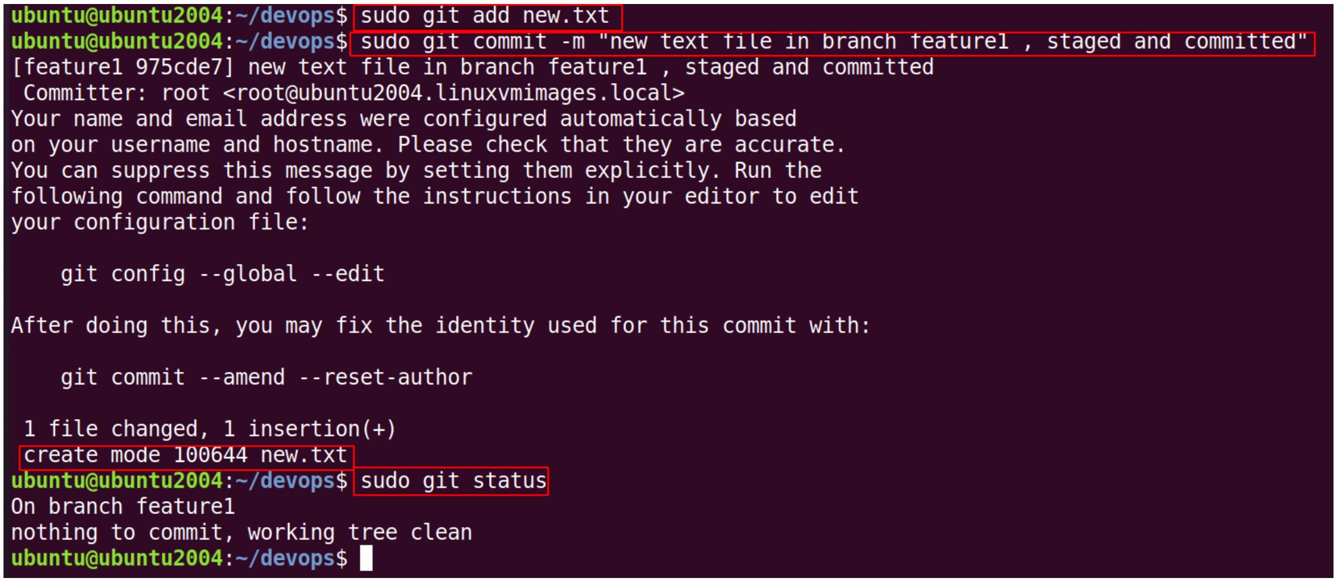
```
ubuntu@ubuntu2004:~/devops$ sudo git branch
* develop
  feature1
  feature2
  master
ubuntu@ubuntu2004:~/devops$ sudo git checkout feature1
Switched to branch 'feature1'
ubuntu@ubuntu2004:~/devops$ sudo git branch
  develop
* feature1
  feature2
  master
```

15. In the branch feature1 we will create a text file called new.txt and add content.



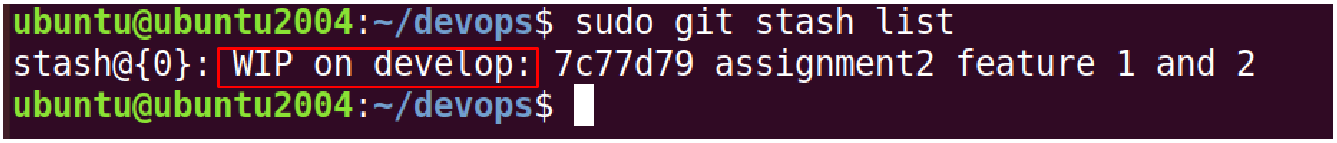
```
ubuntu@ubuntu2004:~/devops$ sudo touch new.txt
ubuntu@ubuntu2004:~/devops$ sudo nano new.txt
ubuntu@ubuntu2004:~/devops$ cat new.txt
This is new text file in the branch feature1
ubuntu@ubuntu2004:~/devops$
```

16. Follow the commands in the screen shot
17. for staging the file new.txt type git add
18. For committing the file type the command git commit -m “comments”



```
ubuntu@ubuntu2004:~/devops$ sudo git add new.txt
ubuntu@ubuntu2004:~/devops$ sudo git commit -m "new text file in branch feature1 , staged and committed"
[feature1 975cde7] new text file in branch feature1 , staged and committed
 Committer: root <root@ubuntu2004.linuxvmimages.local>
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, 1 insertion(+)
 create mode 100644 new.txt
ubuntu@ubuntu2004:~/devops$ sudo git status
On branch feature1
nothing to commit, working tree clean
ubuntu@ubuntu2004:~/devops$
```

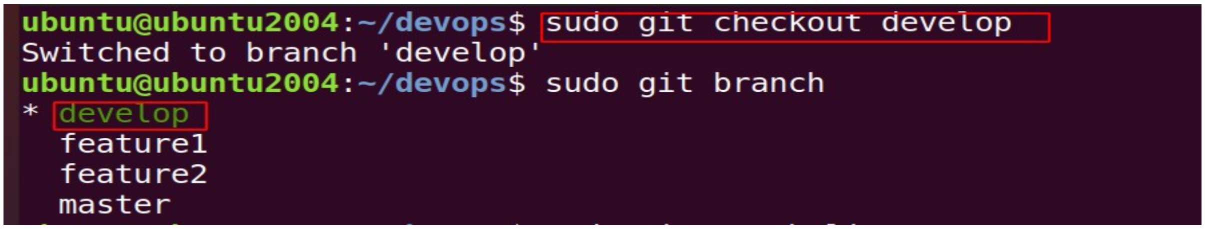
19. Type the command git stash list and you can see on develop branch it shows WIP



```
ubuntu@ubuntu2004:~/devops$ sudo git stash list
stash@{0}: WIP on develop: 7c77d79 assignment2 feature 1 and 2
ubuntu@ubuntu2004:~/devops$
```

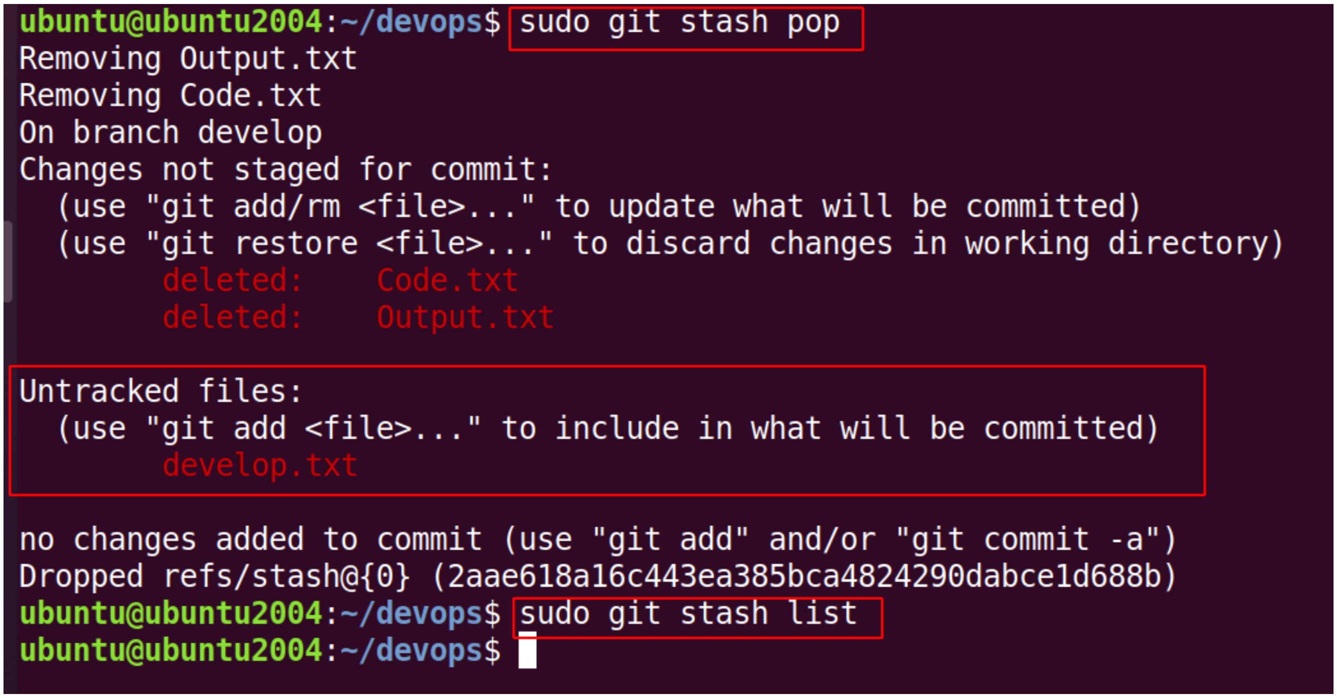
WIP stands for work in progress

20. Now we will checkout to the develop branch



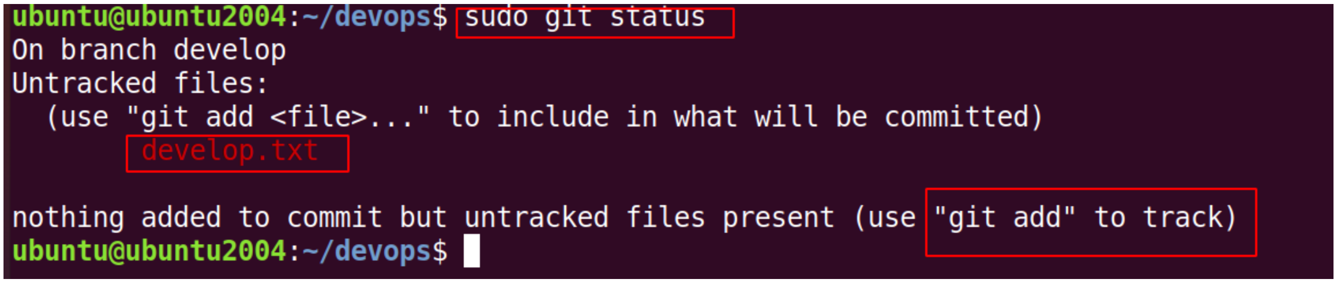
```
ubuntu@ubuntu2004:~/devops$ sudo git checkout develop
Switched to branch 'develop'
ubuntu@ubuntu2004:~/devops$ sudo git branch
* develop
  feature1
  feature2
  master
```

21. Now you want to work on the file in the develop branch and we will use the command git stash pop You can see that now we will be able to stage the file and commit it.



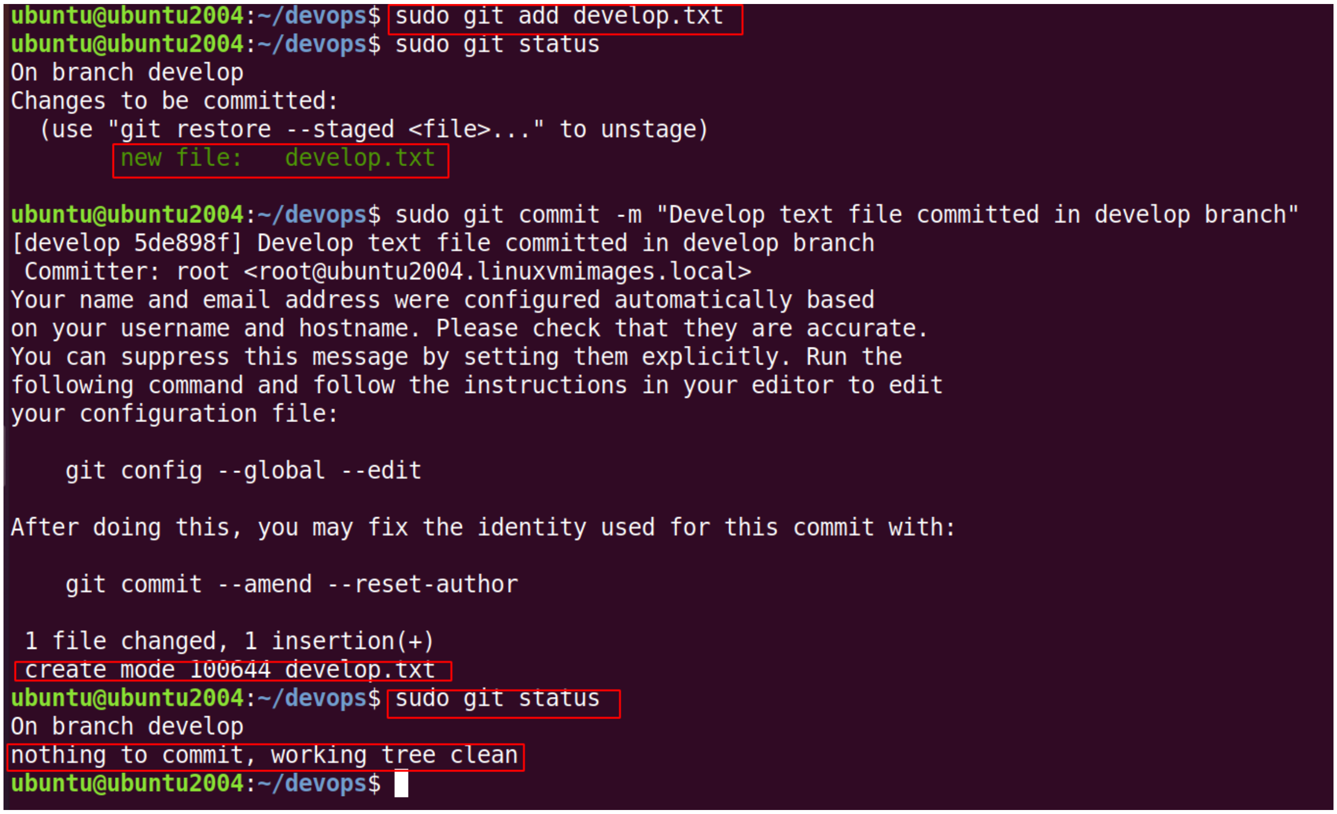
```
ubuntu@ubuntu2004:~/devops$ sudo git stash pop
Removing Output.txt
Removing Code.txt
On branch develop
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    Code.txt
        deleted:    Output.txt

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

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (2aae618a16c443ea385bca4824290dabce1d688b)
ubuntu@ubuntu2004:~/devops$ sudo git stash list
ubuntu@ubuntu2004:~/devops$ 
```

22. Now type the command git status and you can see the develop text file is untracked and can be tracked using the git add command and committed.



```
ubuntu@ubuntu2004:~/devops$ sudo git status
On branch develop
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        develop.txt

nothing added to commit but untracked files present (use "git add" to track)
ubuntu@ubuntu2004:~/devops$ 
```

23. The develop file has been commited.



```
ubuntu@ubuntu2004:~/devops$ sudo git add develop.txt
ubuntu@ubuntu2004:~/devops$ sudo git status
On branch develop
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   develop.txt

ubuntu@ubuntu2004:~/devops$ sudo git commit -m "Develop text file committed in develop branch"
[develop 5de898f] Develop text file committed in develop branch
 Committer: root <root@ubuntu2004.linuxvmimages.local>
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, 1 insertion(+)
 create mode 100644 develop.txt
ubuntu@ubuntu2004:~/devops$ sudo git status
On branch develop
nothing to commit, working tree clean
ubuntu@ubuntu2004:~/devops$ 
```