1 Based on what you have learnt in the class, do the following steps:

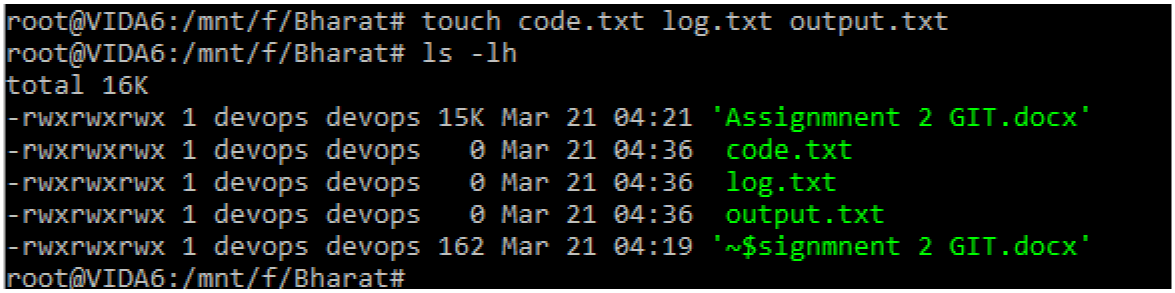
a. Create a new folder

b. b. Put the following files in the folder

Code.txt

Log.txt

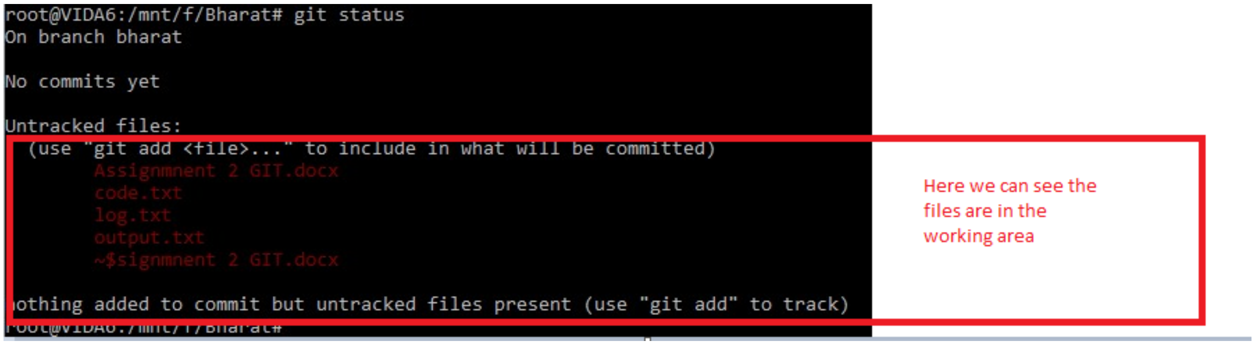
Output.txt



```
root@VIDA6:/mnt/f/Bharat# touch code.txt log.txt output.txt
root@VIDA6:/mnt/f/Bharat# ls -lh
total 16K
-rwxrwxrwx 1 devops devops 15K Mar 21 04:21 'Assignmnent 2 GIT.docx'
-rwxrwxrwx 1 devops devops   0 Mar 21 04:36  code.txt
-rwxrwxrwx 1 devops devops   0 Mar 21 04:36  log.txt
-rwxrwxrwx 1 devops devops   0 Mar 21 04:36  output.txt
-rwxrwxrwx 1 devops devops 162 Mar 21 04:19 '~$signmnent 2 GIT.docx'
root@VIDA6:/mnt/f/Bharat#
```

c. Stage the Code.txt and Output.txt files

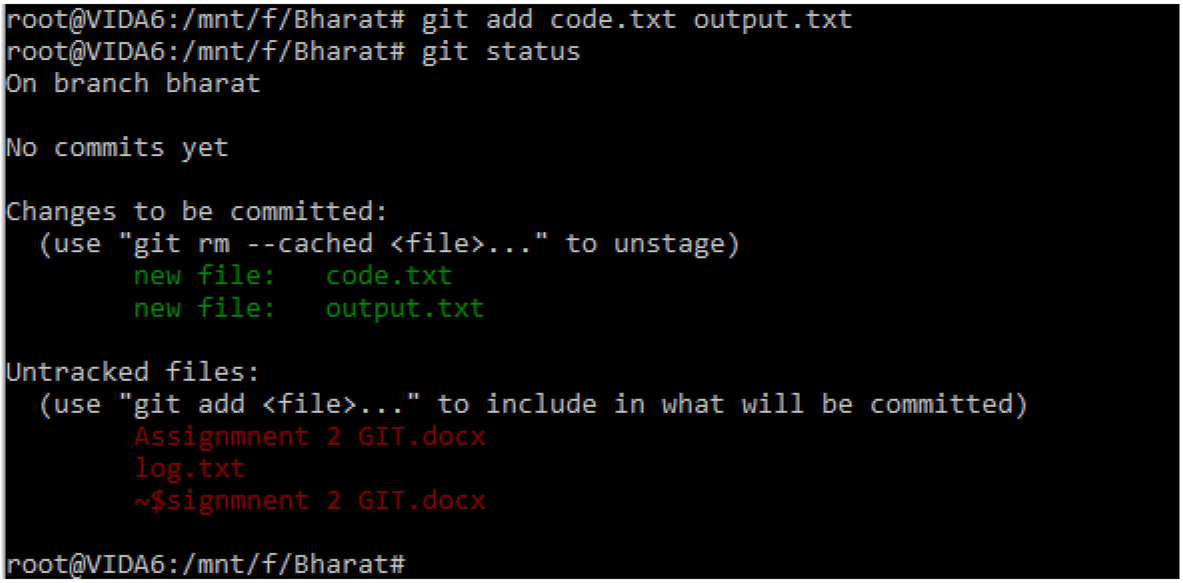
For staging we need to use git add command, using this command in git we can move form working area to staging area. As shown in below command:



```
root@VIDA6:/mnt/f/Bharat# git status
On branch bharat

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Assignmnent 2 GIT.docx
        code.txt
        log.txt
        output.txt
        ~$signmnent 2 GIT.docx

othing added to commit but untracked files present (use "git add" to track)
root@VIDA6:/mnt/f/Bharat#
```

Here we can see the files are in the working area

After then using **git add** code.txt and output.txt

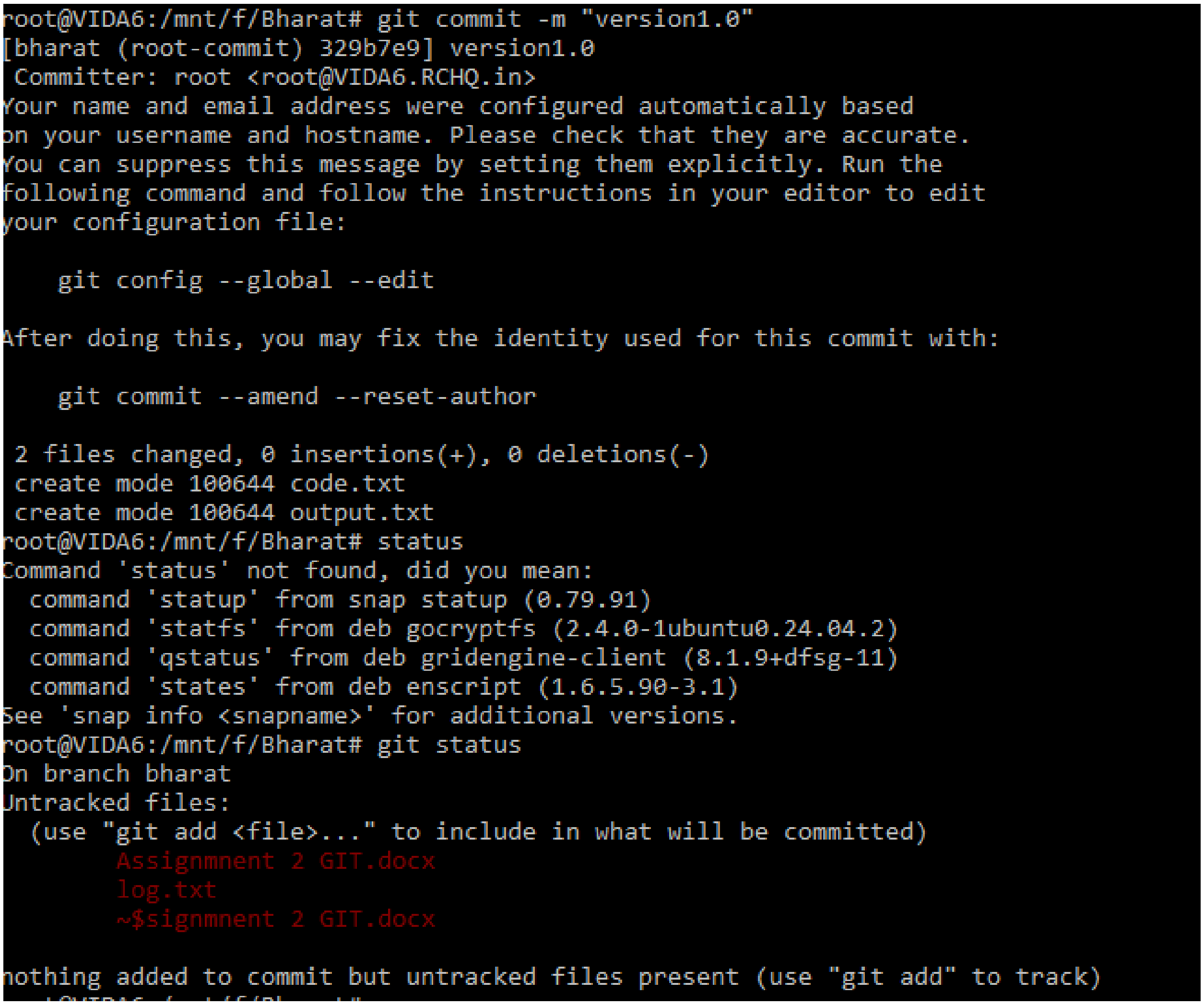


```
root@VIDA6:/mnt/f/Bharat# git add code.txt output.txt
root@VIDA6:/mnt/f/Bharat# git status
On branch bharat

No commits yet

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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Assignmnent 2 GIT.docx
        log.txt
        ~$signmnent 2 GIT.docx

root@VIDA6:/mnt/f/Bharat#
```

d. Commit them

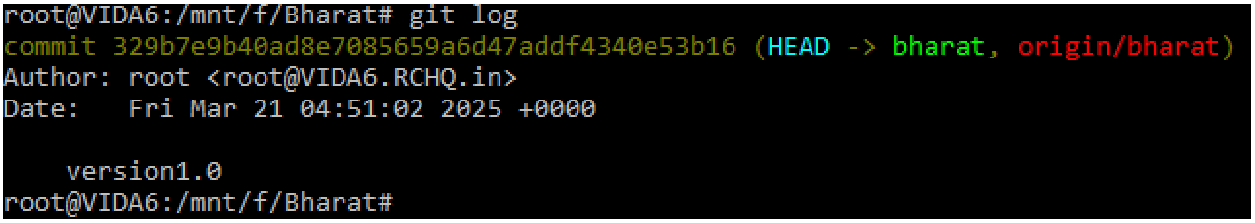
Command used for commit : **git commit -m “commit name”**

We can also call commit at version or check point.



```
root@VIDA6:/mnt/f/Bharat# git commit -m "version1.0"
[bharat (root-commit) 329b7e9] version1.0
 Committer: root <root@VIDA6.RCHQ.in>
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 code.txt
 create mode 100644 output.txt
root@VIDA6:/mnt/f/Bharat# status
Command 'status' not found, did you mean:
  command 'statup' from snap statup (0.79.91)
  command 'statfs' from deb gocryptfs (2.4.0-1ubuntu0.24.04.2)
  command 'qstatus' from deb gridengine-client (8.1.9+dfsg-11)
  command 'states' from deb enscript (1.6.5.90-3.1)
See 'snap info <snapname>' for additional versions.
root@VIDA6:/mnt/f/Bharat# git status
On branch bharat
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Assignmnent 2 GIT.docx
        log.txt
        ~$signmnent 2 GIT.docx

nothing added to commit but untracked files present (use "git add" to track)
```

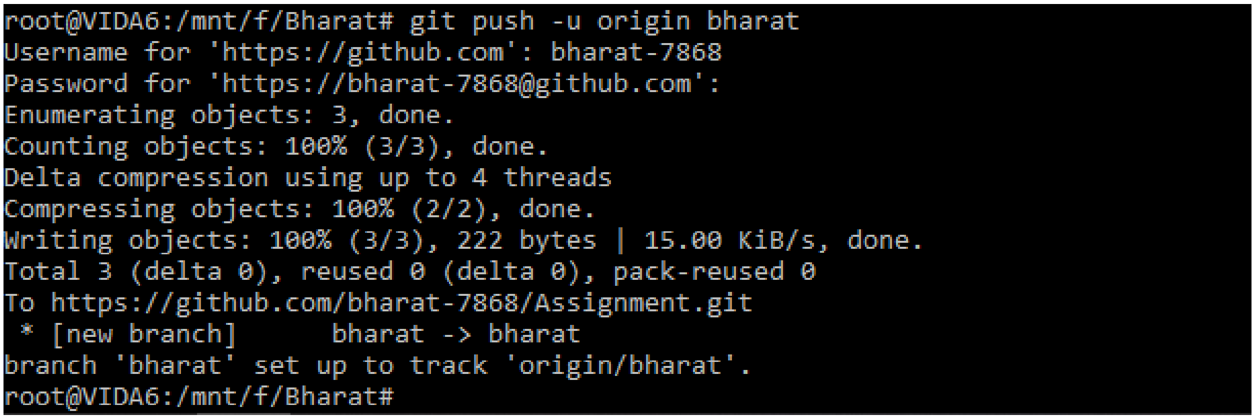
From git log we can check the commit details



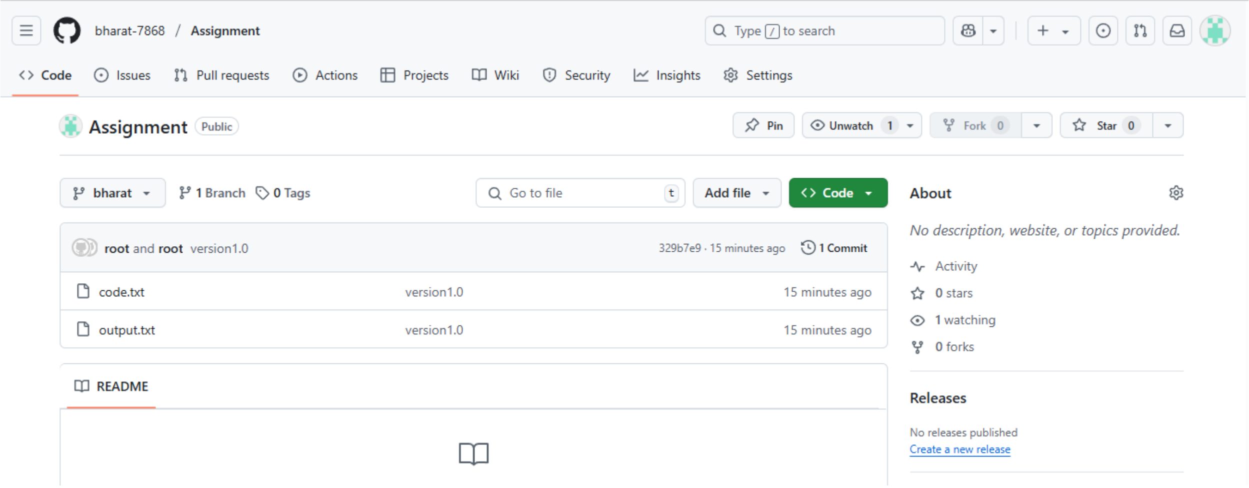
```
root@VIDA6:/mnt/f/Bharat# git log
commit 329b7e9b40ad8e7085659a6d47addf4340e53b16 (HEAD -> bharat, origin/bharat)
Author: root <root@VIDA6.RCHQ.in>
Date:   Fri Mar 21 04:51:02 2025 +0000

    version1.0
root@VIDA6:/mnt/f/Bharat#
```

E: And finally push them to GitHub

For push any data to git hub repository **git push -u origin brnachname**



```
root@VIDA6:/mnt/f/Bharat# git push -u origin bharat
Username for 'https://github.com': bharat-7868
Password for 'https://bharat-7868@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 222 bytes | 15.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/bharat-7868/Assignment.git
 * [new branch]      bharat -> bharat
branch 'bharat' set up to track 'origin/bharat'.
root@VIDA6:/mnt/f/Bharat#
```

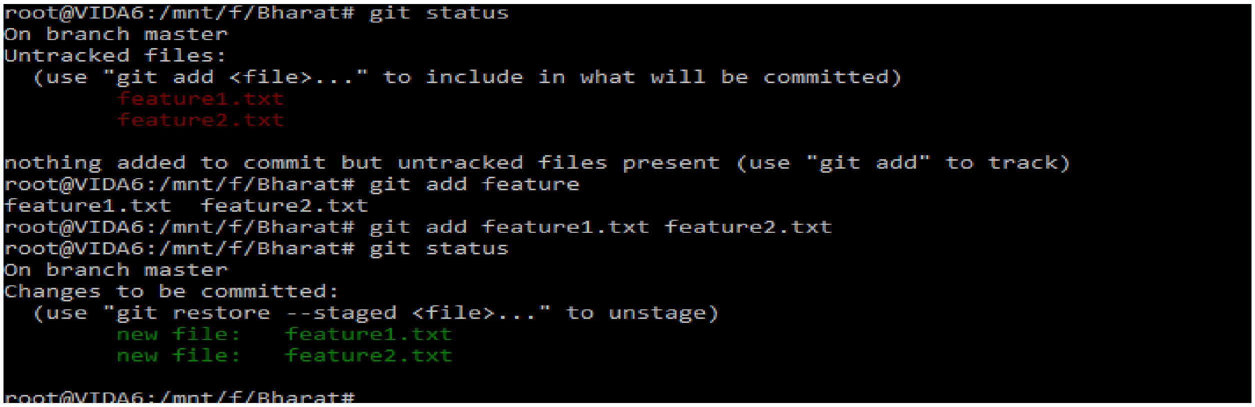


root and root version1.0 329b7e9 · 15 minutes ago 1 Commit

code.txt	version1.0	15 minutes ago
output.txt	version1.0	15 minutes ago

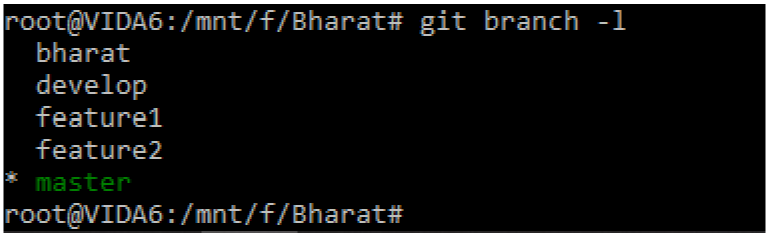
Q 2: Tasks to Be Performed:

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



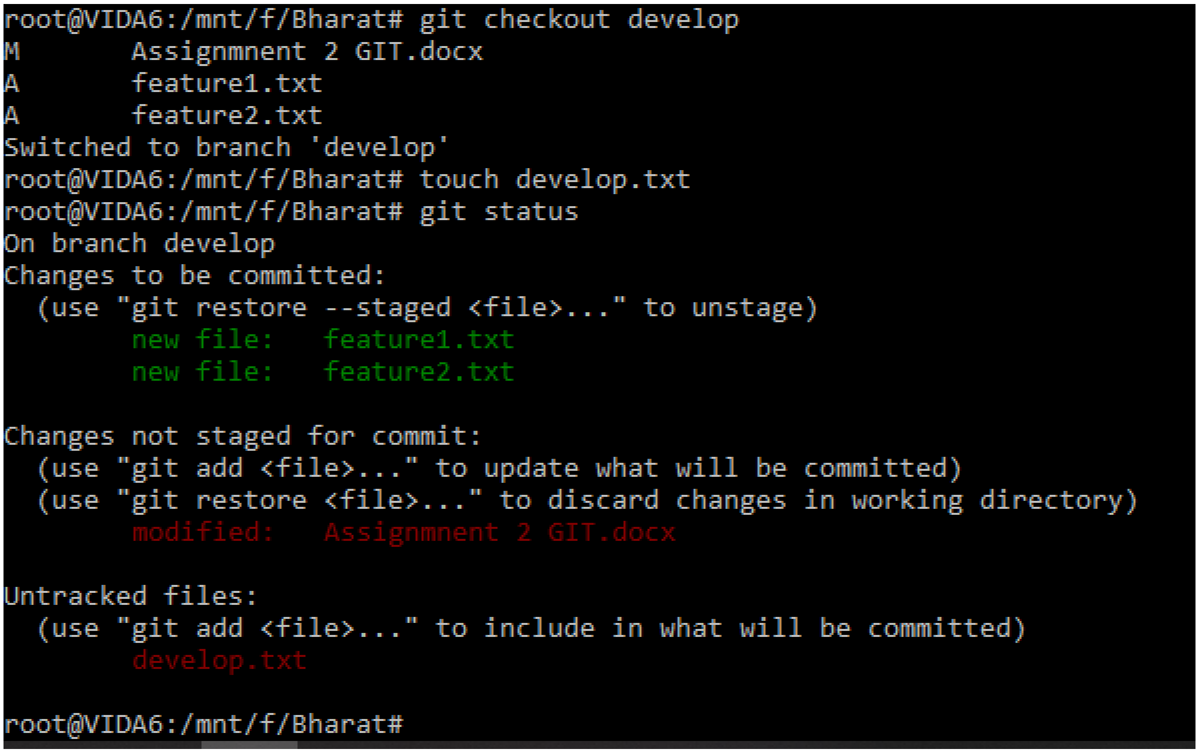
```
root@VIDA6:/mnt/f/Bharat# git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        feature1.txt
        feature2.txt

nothing added to commit but untracked files present (use "git add" to track)
root@VIDA6:/mnt/f/Bharat# git add feature
feature1.txt  feature2.txt
root@VIDA6:/mnt/f/Bharat# git add feature1.txt feature2.txt
root@VIDA6:/mnt/f/Bharat# git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   feature1.txt
        new file:   feature2.txt

root@VIDA6:/mnt/f/Bharat#
```

2. Create 3 branches develop, feature1 and feature2



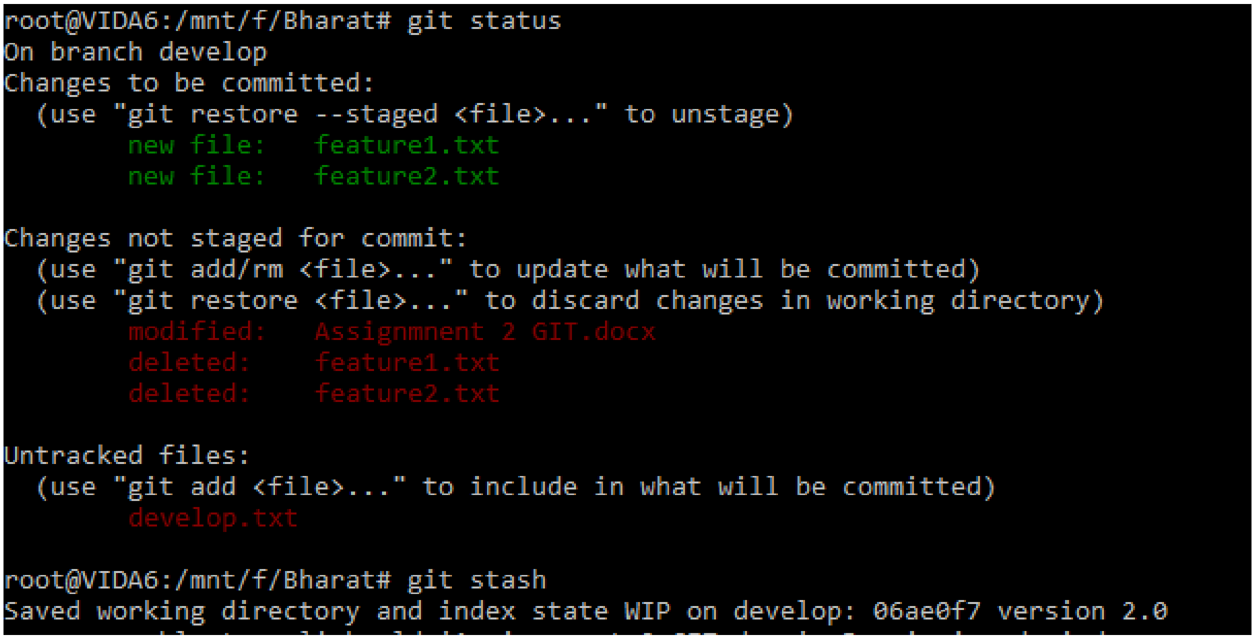
```
root@VIDA6:/mnt/f/Bharat# git branch -l
  bharat
  develop
  feature1
  feature2
* master
root@VIDA6:/mnt/f/Bharat#
```

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



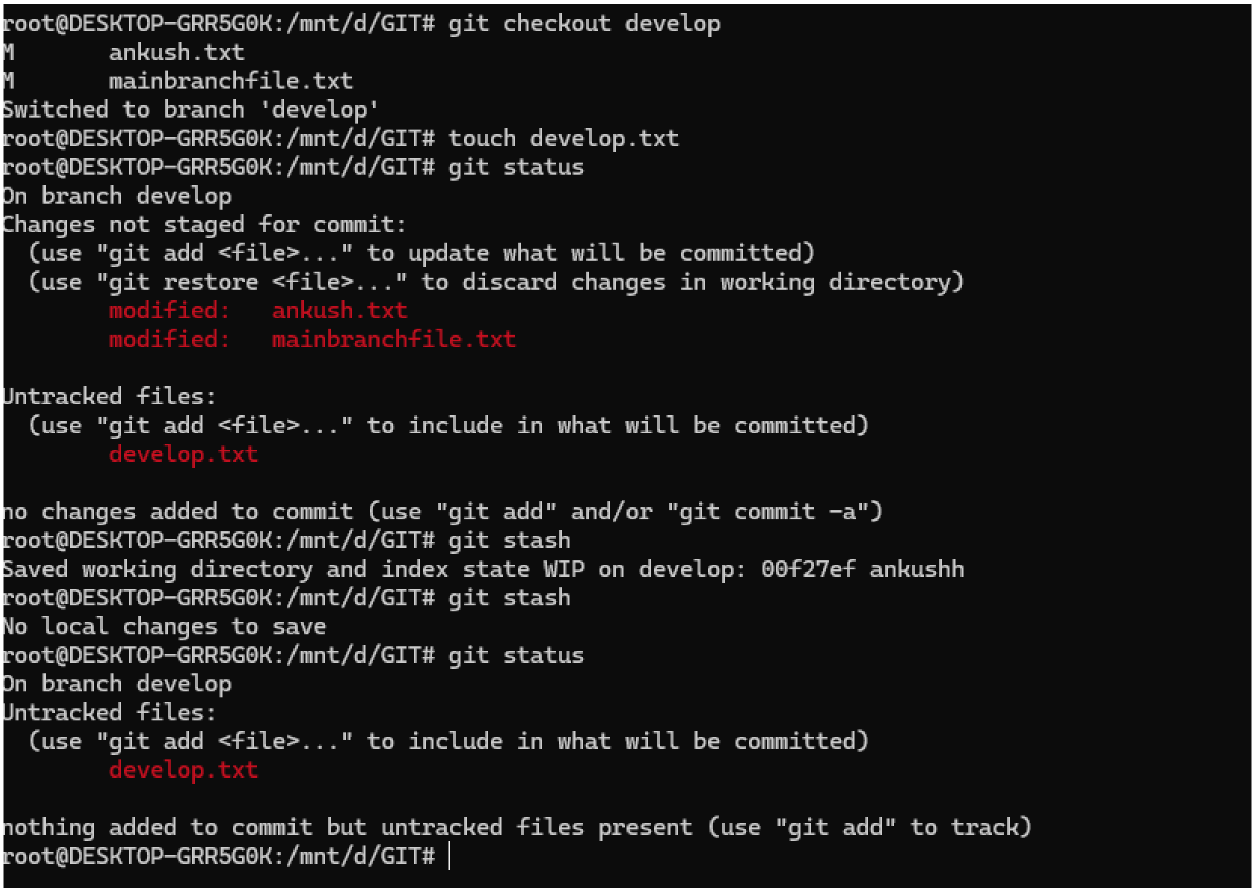
```
root@VIDA6:/mnt/f/Bharat# git checkout develop
M       Assignmnent 2 GIT.docx
A       feature1.txt
A       feature2.txt
Switched to branch 'develop'
root@VIDA6:/mnt/f/Bharat# touch develop.txt
root@VIDA6:/mnt/f/Bharat# git status
On branch develop
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   feature1.txt
        new file:   feature2.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Assignmnent 2 GIT.docx

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

root@VIDA6:/mnt/f/Bharat#
```

4. Stash this file and check out to feature1 branch



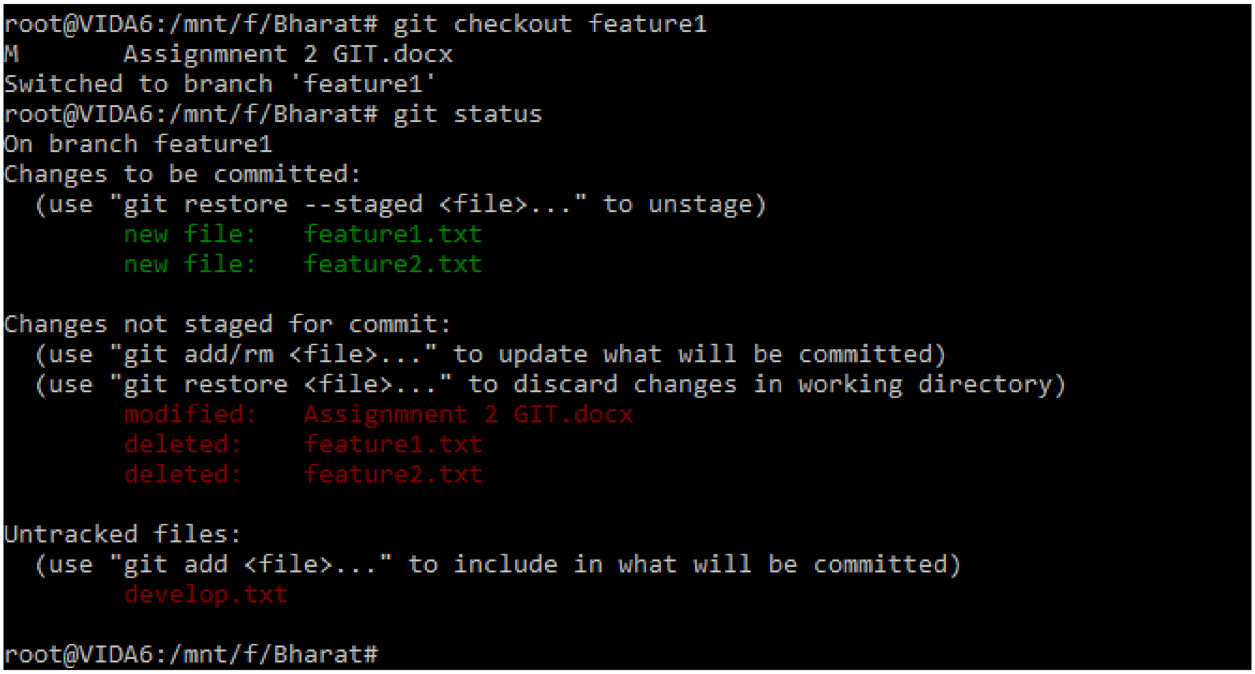
```
root@VIDA6:/mnt/f/Bharat# git status
On branch develop
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   feature1.txt
        new file:   feature2.txt

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)
        modified:   Assignmnent 2 GIT.docx
        deleted:    feature1.txt
        deleted:    feature2.txt

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

root@VIDA6:/mnt/f/Bharat# git stash
Saved working directory and index state WIP on develop: 06ae0f7 version 2.0
```

Now we will stash the file as shown in below image



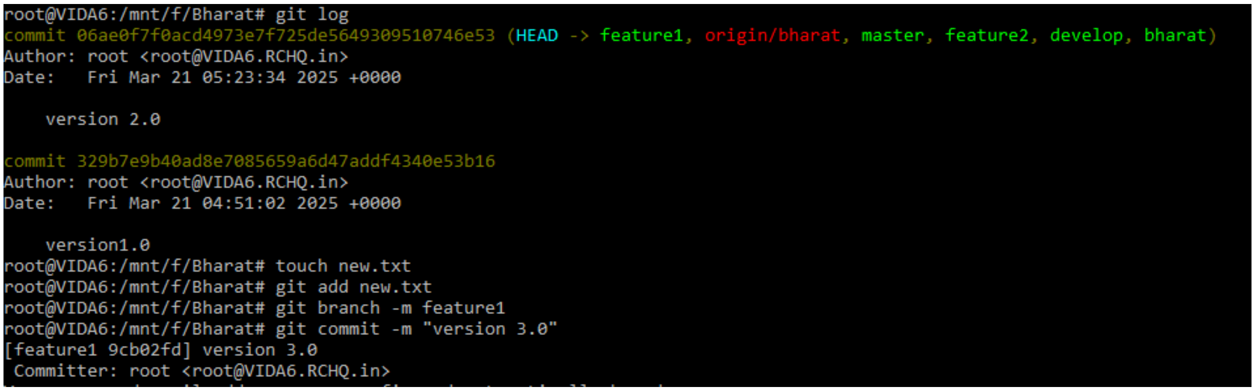
```
root@DESKTOP-GRR5G0K:/mnt/d/GIT# git checkout develop
M       ankush.txt
M       mainbranchfile.txt
Switched to branch 'develop'
root@DESKTOP-GRR5G0K:/mnt/d/GIT# touch develop.txt
root@DESKTOP-GRR5G0K:/mnt/d/GIT# git status
On branch develop
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   ankush.txt
        modified:   mainbranchfile.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")
root@DESKTOP-GRR5G0K:/mnt/d/GIT# git stash
Saved working directory and index state WIP on develop: 00f27ef ankushh
root@DESKTOP-GRR5G0K:/mnt/d/GIT# git stash
No local changes to save
root@DESKTOP-GRR5G0K:/mnt/d/GIT# 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)
root@DESKTOP-GRR5G0K:/mnt/d/GIT#
```

After then we will switch to feature1 branch.



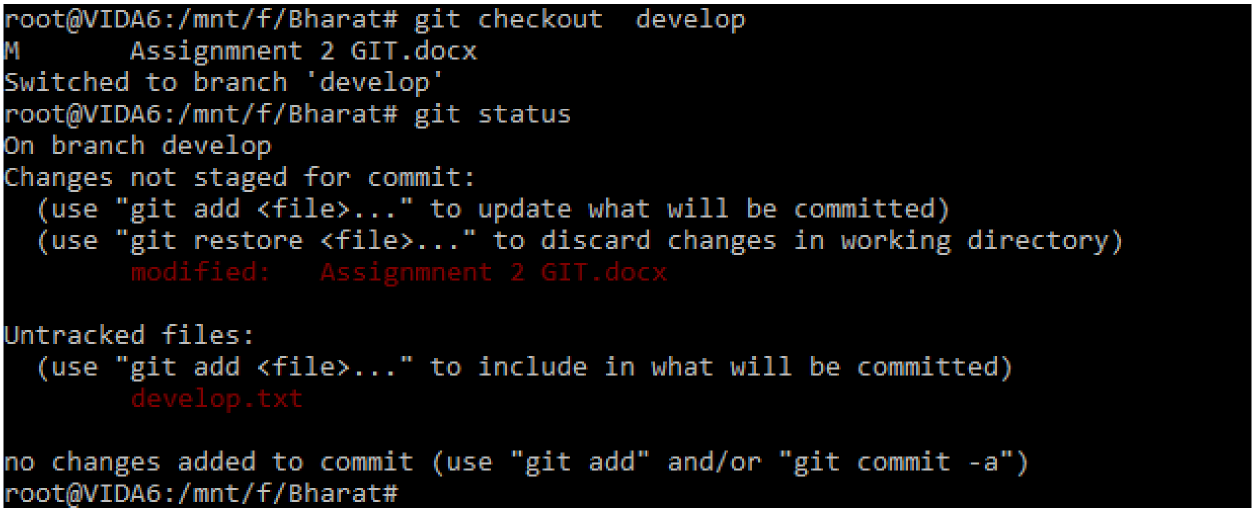
```
root@VIDA6:/mnt/f/Bharat# git checkout feature1
M       Assignmnent 2 GIT.docx
Switched to branch 'feature1'
root@VIDA6:/mnt/f/Bharat# git status
On branch feature1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   feature1.txt
        new file:   feature2.txt

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)
        modified:   Assignmnent 2 GIT.docx
        deleted:    feature1.txt
        deleted:    feature2.txt

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

root@VIDA6:/mnt/f/Bharat#
```

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



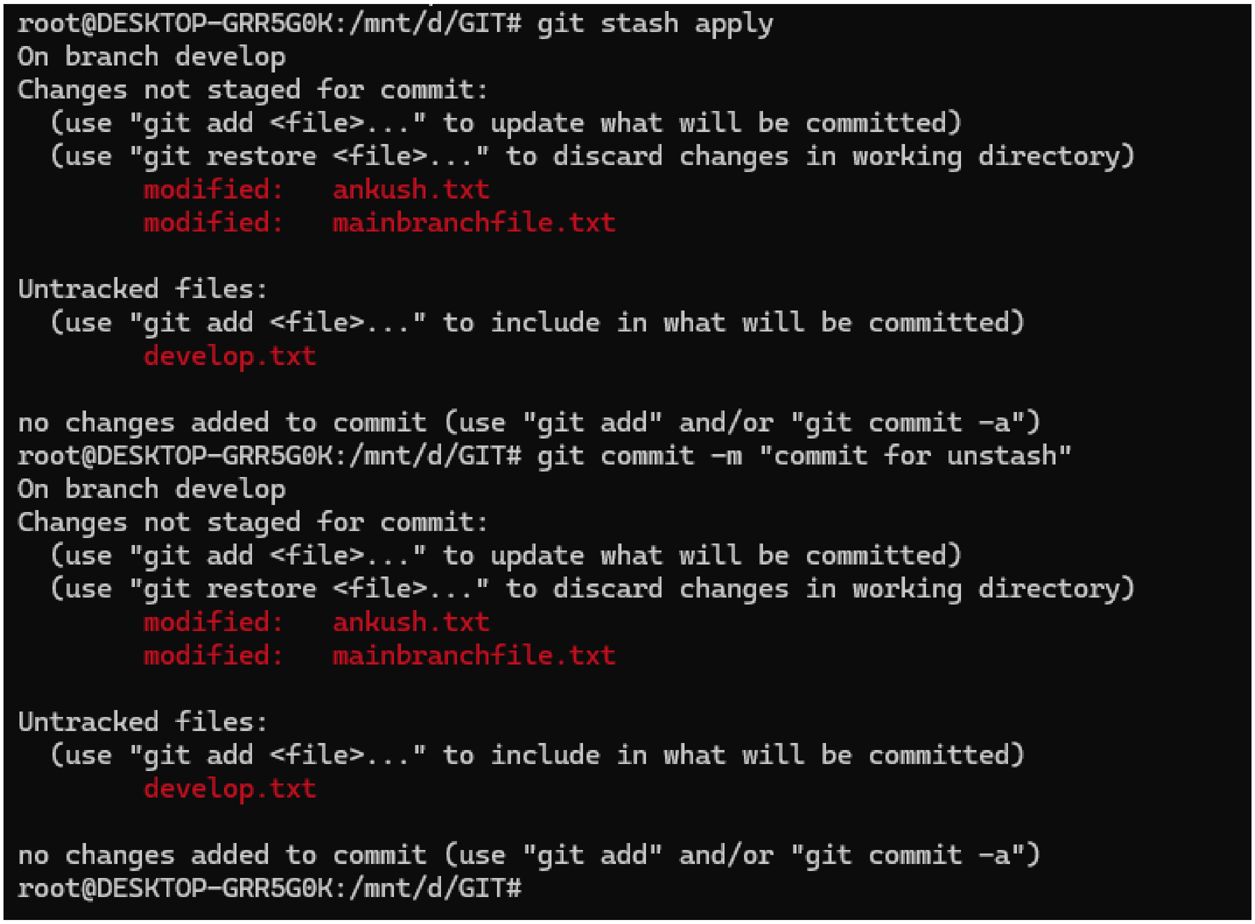
```
root@VIDA6:/mnt/f/Bharat# git log
commit 06ae0f7f0acd4973e7f725de5649309510746e53 (HEAD -> feature1, origin/bharat, master, feature2, develop, bharat)
Author: root <root@VIDA6.RCHQ.in>
Date:   Fri Mar 21 05:23:34 2025 +0000

    version 2.0

commit 329b7e9b40ad8e7085659a6d47addf4340e53b16
Author: root <root@VIDA6.RCHQ.in>
Date:   Fri Mar 21 04:51:02 2025 +0000

    version1.0
root@VIDA6:/mnt/f/Bharat# touch new.txt
root@VIDA6:/mnt/f/Bharat# git add new.txt
root@VIDA6:/mnt/f/Bharat# git branch -m feature1
root@VIDA6:/mnt/f/Bharat# git commit -m "version 3.0"
[feature1 9cb02fd] version 3.0
 Committer: root <root@VIDA6.RCHQ.in>
```

6. Checkout to develop, unstash this file and commit



```
root@VIDA6:/mnt/f/Bharat# git checkout  develop
M       Assignmnent 2 GIT.docx
Switched to branch 'develop'
root@VIDA6:/mnt/f/Bharat# git status
On branch develop
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Assignmnent 2 GIT.docx

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")
root@VIDA6:/mnt/f/Bharat#
```

For unstash the file we use command “**git stash apply**” as shown in below image

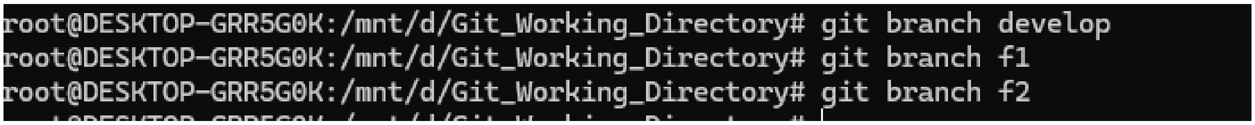


```
root@DESKTOP-GRR5G0K:/mnt/d/GIT# git stash apply
On branch develop
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   ankush.txt
        modified:   mainbranchfile.txt

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")
root@DESKTOP-GRR5G0K:/mnt/d/GIT# git commit -m "commit for unstash"
On branch develop
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   ankush.txt
        modified:   mainbranchfile.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")
root@DESKTOP-GRR5G0K:/mnt/d/GIT#
```

7. Please submit all the Git commands used to do the above step>
   You can check all the commands in the screen shots it self.

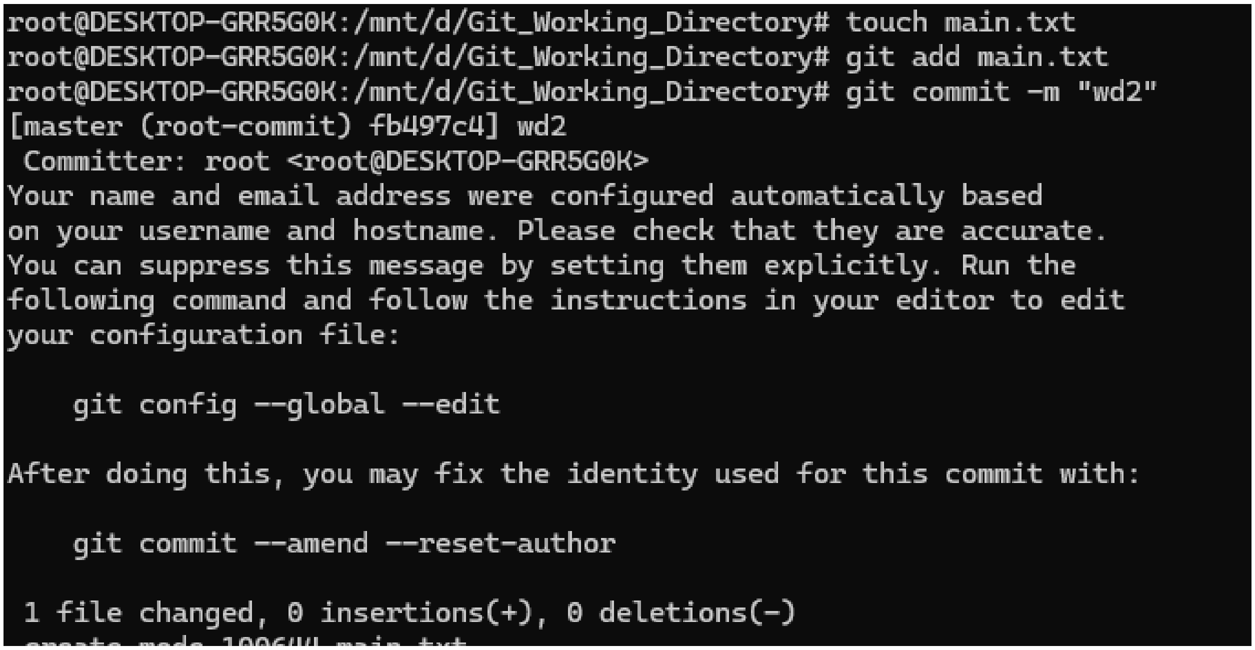
Q3. Tasks to be performed.

1. Create a Git working directory, with the following branches: Develop F1 f2



```
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory# git branch develop
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory# git branch f1
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory# git branch f2
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory#
```

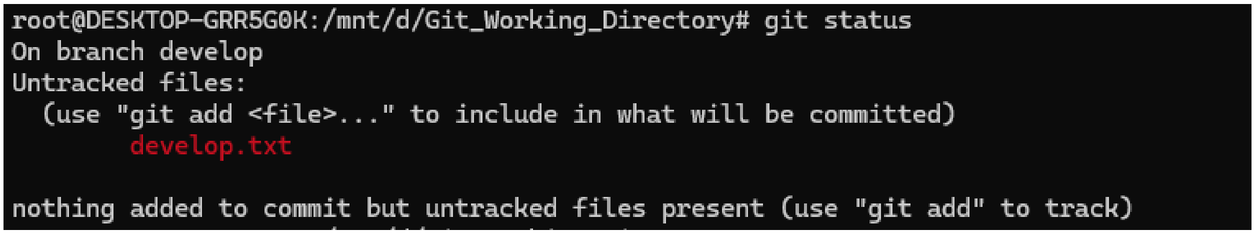
2. In the master branch commit main.txt as shown below



```
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory# touch main.txt
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory# git add main.txt
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory# git commit -m "wd2"
[master (root-commit) fb497c4] wd2
 Committer: root <root@DESKTOP-GRR5G0K>
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, 0 insertions(+), 0 deletions(-)
 create mode 100644 main.txt
```

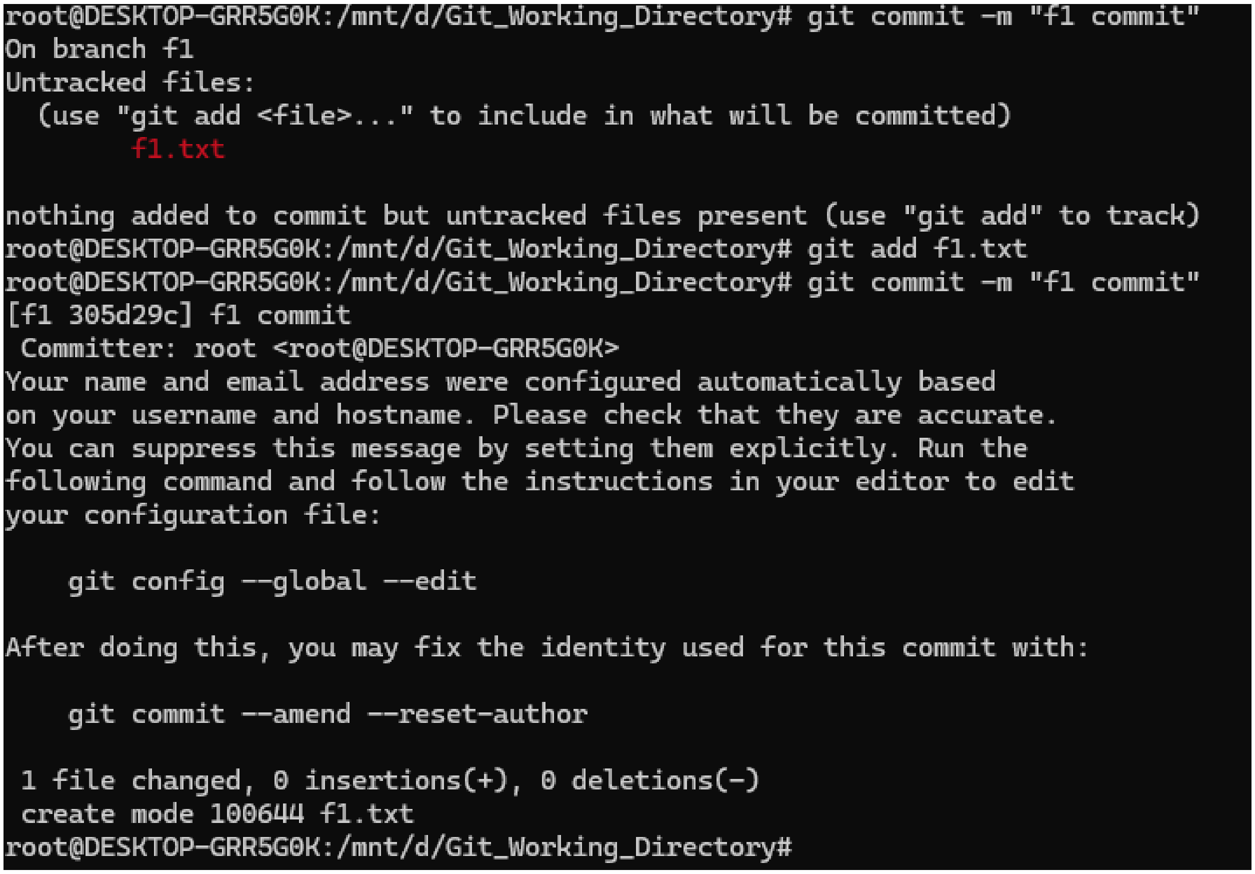
3. Put develop.txt in develop branch, f1.txt and f2.txt in f1 and f2 respectively

For develop branch



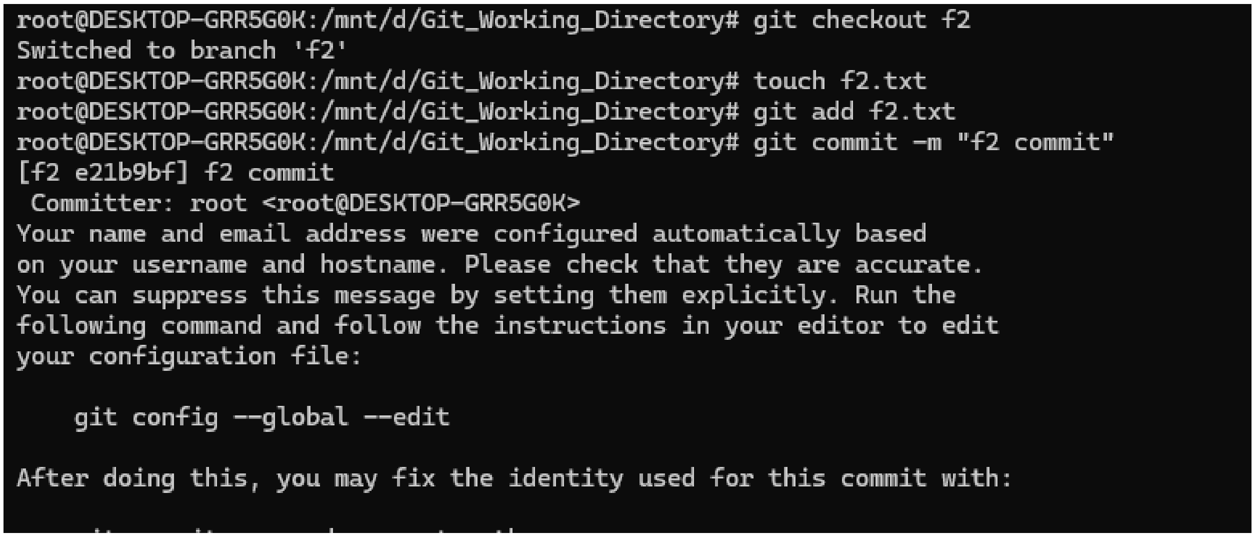
```
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory# git status
On branch develop
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        develop.txt

nothing added to commit but untracked files present (use "git add" to track)
```

For f1 brnach



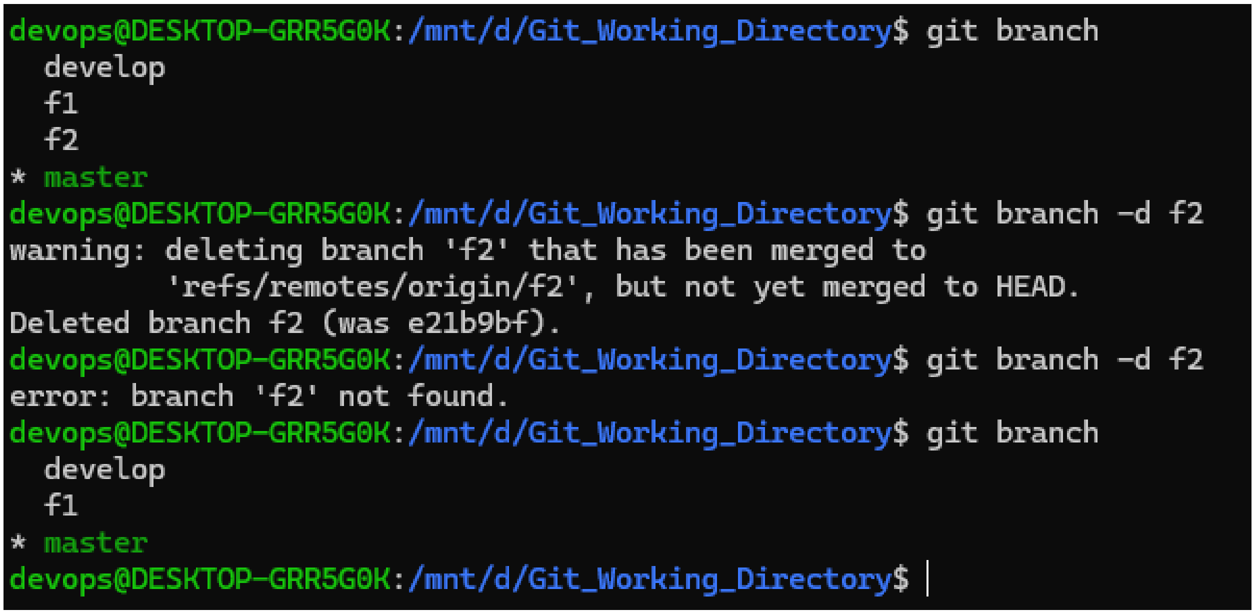
```
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory# git commit -m "f1 commit"
On branch f1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        f1.txt

nothing added to commit but untracked files present (use "git add" to track)
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory# git add f1.txt
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory# git commit -m "f1 commit"
[f1 305d29c] f1 commit
 Committer: root <root@DESKTOP-GRR5G0K>
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, 0 insertions(+), 0 deletions(-)
 create mode 100644 f1.txt
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory#
```

For f2 brnach

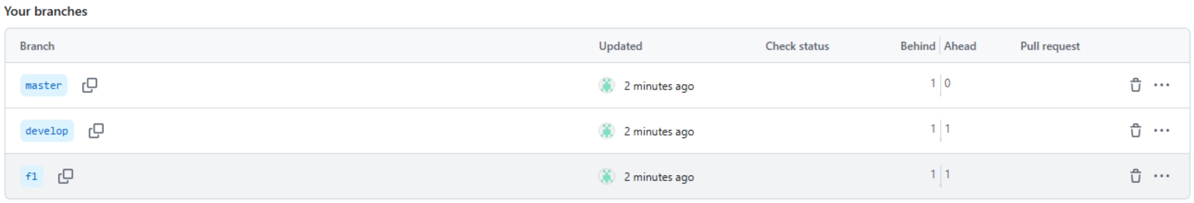


```
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory# git checkout f2
Switched to branch 'f2'
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory# touch f2.txt
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory# git add f2.txt
root@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory# git commit -m "f2 commit"
[f2 e21b9bf] f2 commit
 Committer: root <root@DESKTOP-GRR5G0K>
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:
```

Now we will push all branches to git hub repository Also we have deleted f2 brnach



```
devops@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory$ git branch
  develop
  f1
  f2
* master
devops@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory$ git branch -d f2
warning: deleting branch 'f2' that has been merged to
         'refs/remotes/origin/f2', but not yet merged to HEAD.
Deleted branch f2 (was e21b9bf).
devops@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory$ git branch -d f2
error: branch 'f2' not found.
devops@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory$ git branch
  develop
  f1
* master
devops@DESKTOP-GRR5G0K:/mnt/d/Git_Working_Directory$ 
```

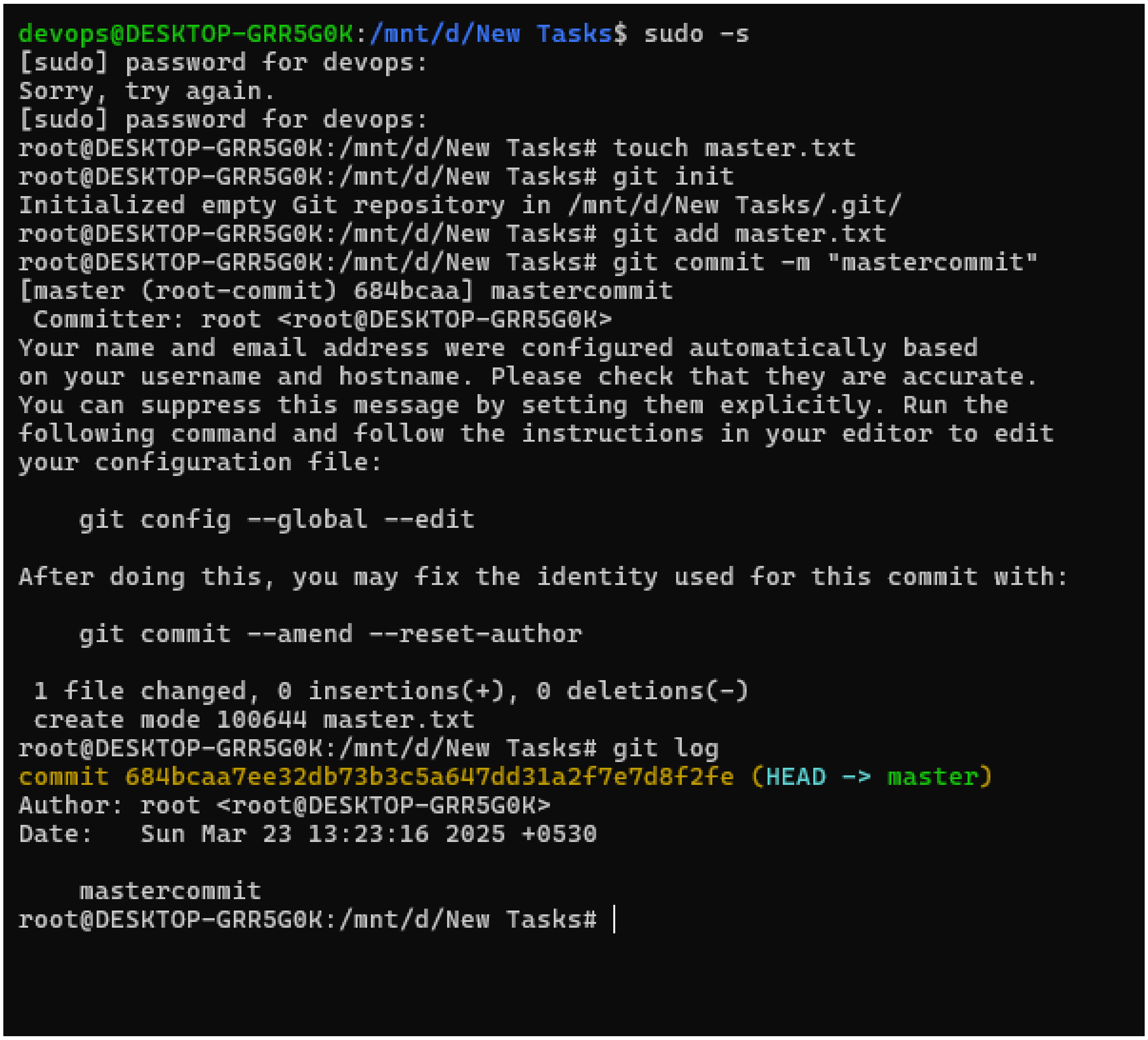


Your branches

Branch	Updated	Check status	Behind	Ahead	Pull request
master	2 minutes ago		1	0	
develop	2 minutes ago		1	1	
f1	2 minutes ago		1	1	

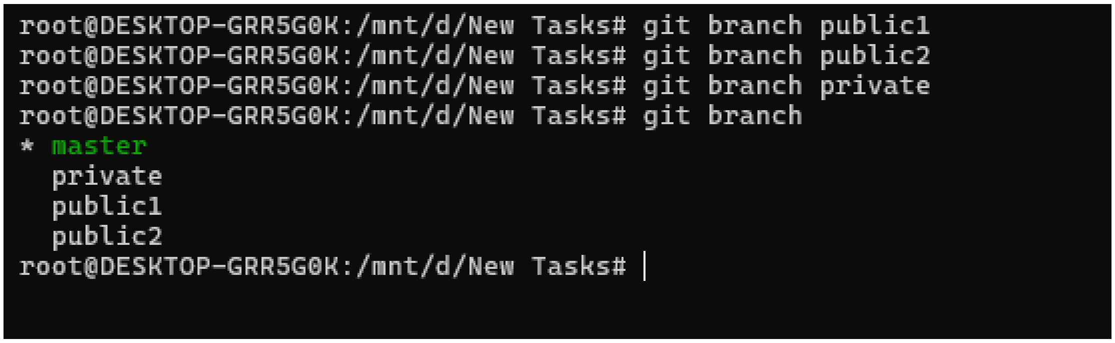
Q4 Tasks to Be Performed:

1. Put master.txt on master branch, stage and commit



```
devops@DESKTOP-GRR5G0K:/mnt/d/New Tasks$ sudo -s
[sudo] password for devops:
Sorry, try again.
[sudo] password for devops:
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# touch master.txt
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git init
Initialized empty Git repository in /mnt/d/New Tasks/.git/
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git add master.txt
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git commit -m "mastercommit"
[master (root-commit) 684bcaa] mastercommit
 Committer: root <root@DESKTOP-GRR5G0K>
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, 0 insertions(+), 0 deletions(-)
 create mode 100644 master.txt
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git log
commit 684bcaa7ee32db73b3c5a647dd31a2f7e7d8f2fe (HEAD -> master)
Author: root <root@DESKTOP-GRR5G0K>
Date:   Sun Mar 23 13:23:16 2025 +0530

    mastercommit
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# |
```

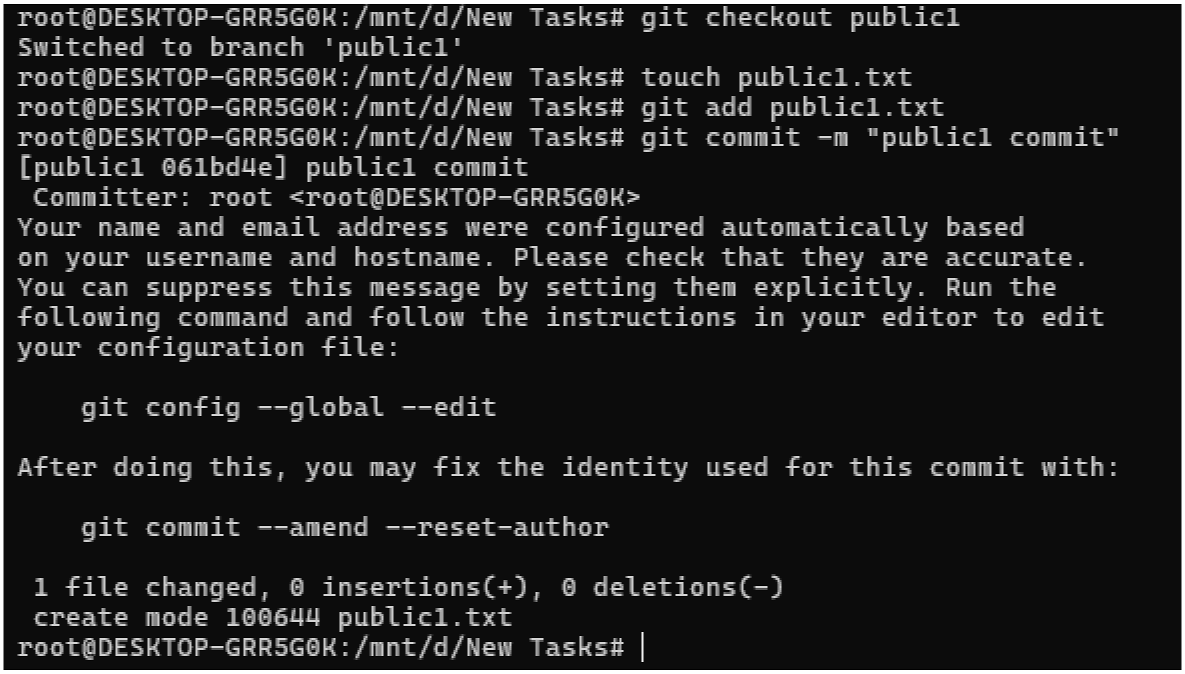
2 Create 3 branches: public 1, public 2 and private



```
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git branch public1
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git branch public2
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git branch private
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git branch
* master
  private
  public1
  public2
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# |
```

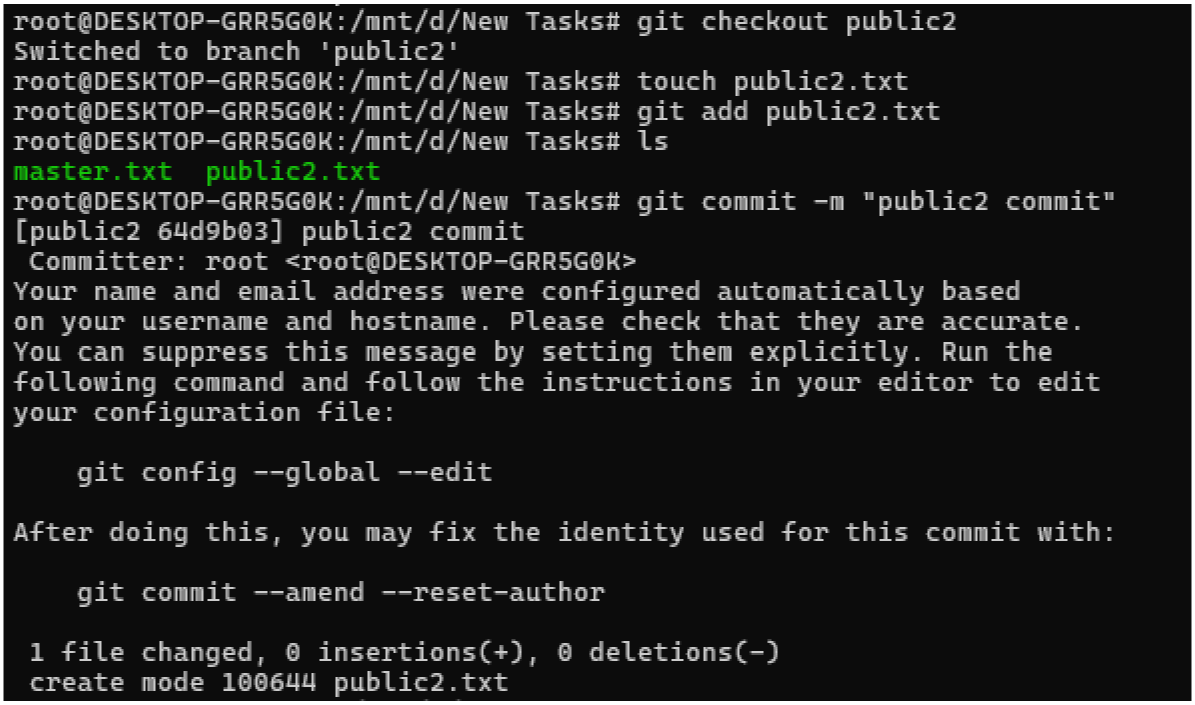
3 Put public1.txt on public 1 branch, stage and commit

For public1



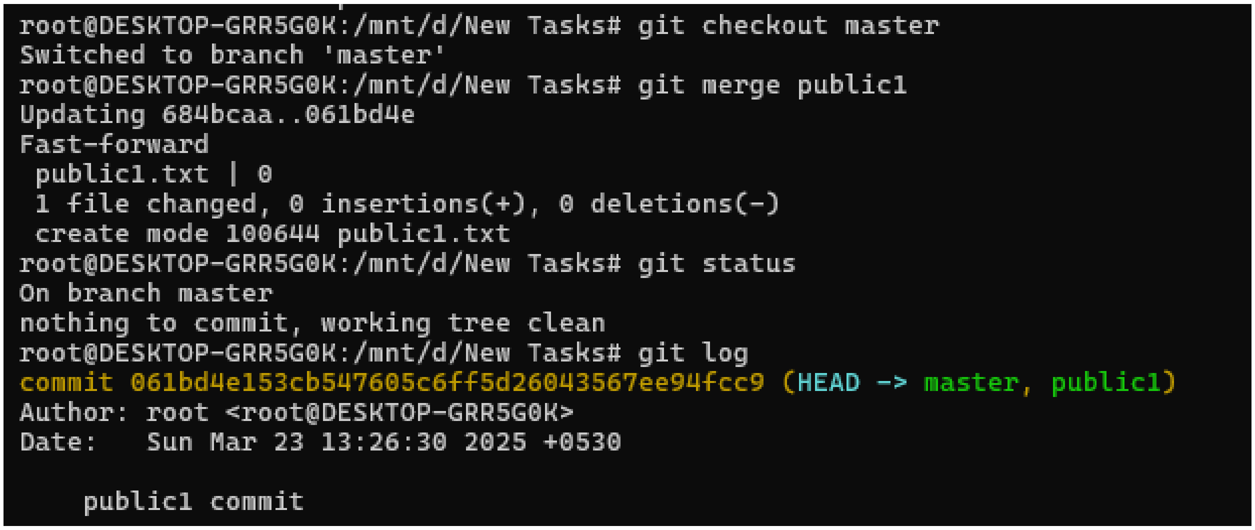
```
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git checkout public1
Switched to branch 'public1'
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# touch public1.txt
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git add public1.txt
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git commit -m "public1 commit"
[public1 061bd4e] public1 commit
 Committer: root <root@DESKTOP-GRR5G0K>
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, 0 insertions(+), 0 deletions(-)
 create mode 100644 public1.txt
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# |
```

For public2



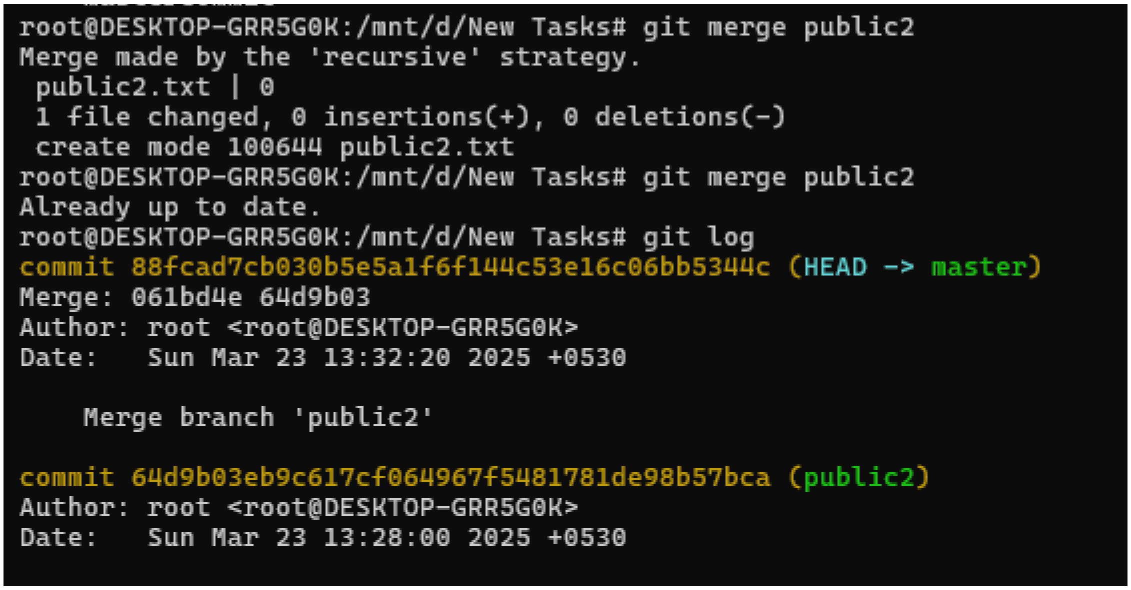
```
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git checkout public2
Switched to branch 'public2'
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# touch public2.txt
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git add public2.txt
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# ls
master.txt  public2.txt
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git commit -m "public2 commit"
[public2 64d9b03] public2 commit
 Committer: root <root@DESKTOP-GRR5G0K>
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, 0 insertions(+), 0 deletions(-)
 create mode 100644 public2.txt
```

4 Merge public 1 on master branch



```
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git checkout master
Switched to branch 'master'
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git merge public1
Updating 684bcaa..061bd4e
Fast-forward
 public1.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 public1.txt
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git status
On branch master
nothing to commit, working tree clean
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git log
commit 061bd4e153cb547605c6ff5d26043567ee94fcc9 (HEAD -> master, public1)
Author: root <root@DESKTOP-GRR5G0K>
Date:   Sun Mar 23 13:26:30 2025 +0530

    public1 commit
```

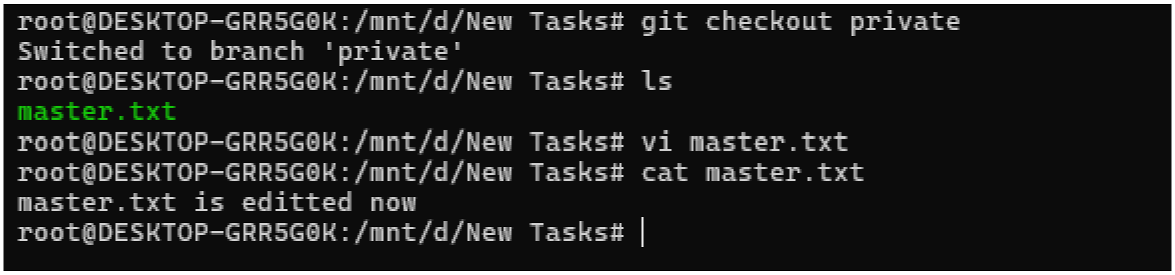
5 Merge public 2 on master branch



```
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git merge public2
Merge made by the 'recursive' strategy.
 public2.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 public2.txt
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git merge public2
Already up to date.
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git log
commit 88fcad7cb030b5e5a1f6f144c53e16c06bb5344c (HEAD -> master)
Merge: 061bd4e 64d9b03
Author: root <root@DESKTOP-GRR5G0K>
Date:   Sun Mar 23 13:32:20 2025 +0530

    Merge branch 'public2'

commit 64d9b03eb9c617cf064967f5481781de98b57bca (public2)
Author: root <root@DESKTOP-GRR5G0K>
Date:   Sun Mar 23 13:28:00 2025 +0530
```

6 Edit master.txt on private branch, stage and commit

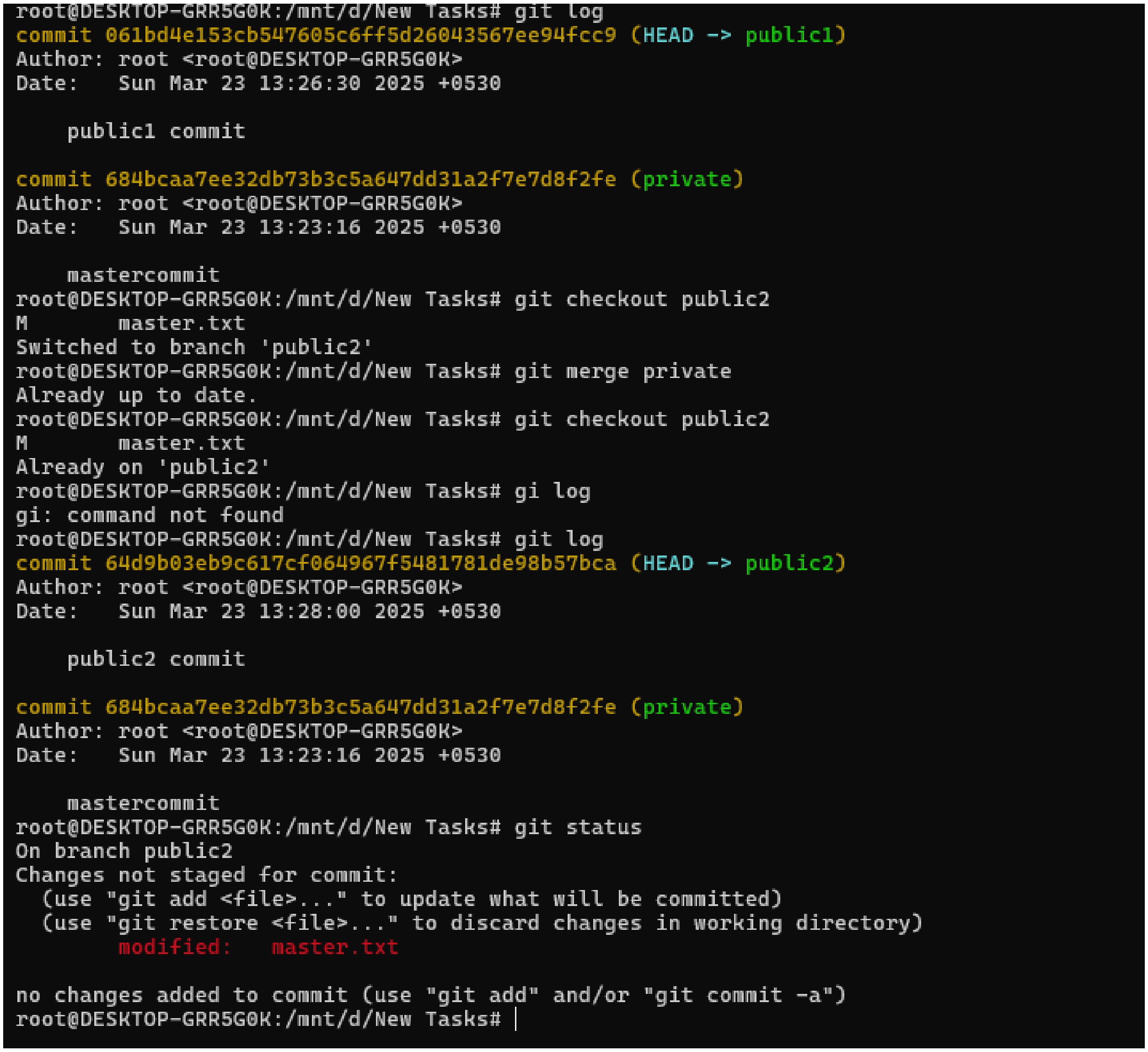
For this now we will switch to private branch and edit the file



```
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git checkout private
Switched to branch 'private'
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# ls
master.txt
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# vi master.txt
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# cat master.txt
master.txt is editted now
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# |
```

7 Now update branch public 1 and public 2 with new master code in private

For this we will merge private branch in both public1 and public2



```
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git log
commit 061bd4e153cb547605c6ff5d26043567ee94fcc9 (HEAD -> public1)
Author: root <root@DESKTOP-GRR5G0K>
Date:   Sun Mar 23 13:26:30 2025 +0530

    public1 commit

commit 684bcaa7ee32db73b3c5a647dd31a2f7e7d8f2fe (private)
Author: root <root@DESKTOP-GRR5G0K>
Date:   Sun Mar 23 13:23:16 2025 +0530

    mastercommit
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git checkout public2
M       master.txt
Switched to branch 'public2'
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git merge private
Already up to date.
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git checkout public2
M       master.txt
Already on 'public2'
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# gi log
gi: command not found
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git log
commit 64d9b03eb9c617cf064967f5481781de98b57bca (HEAD -> public2)
Author: root <root@DESKTOP-GRR5G0K>
Date:   Sun Mar 23 13:28:00 2025 +0530

    public2 commit

commit 684bcaa7ee32db73b3c5a647dd31a2f7e7d8f2fe (private)
Author: root <root@DESKTOP-GRR5G0K>
Date:   Sun Mar 23 13:23:16 2025 +0530

    mastercommit
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git status
On branch public2
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   master.txt

no changes added to commit (use "git add" and/or "git commit -a")
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# 
```

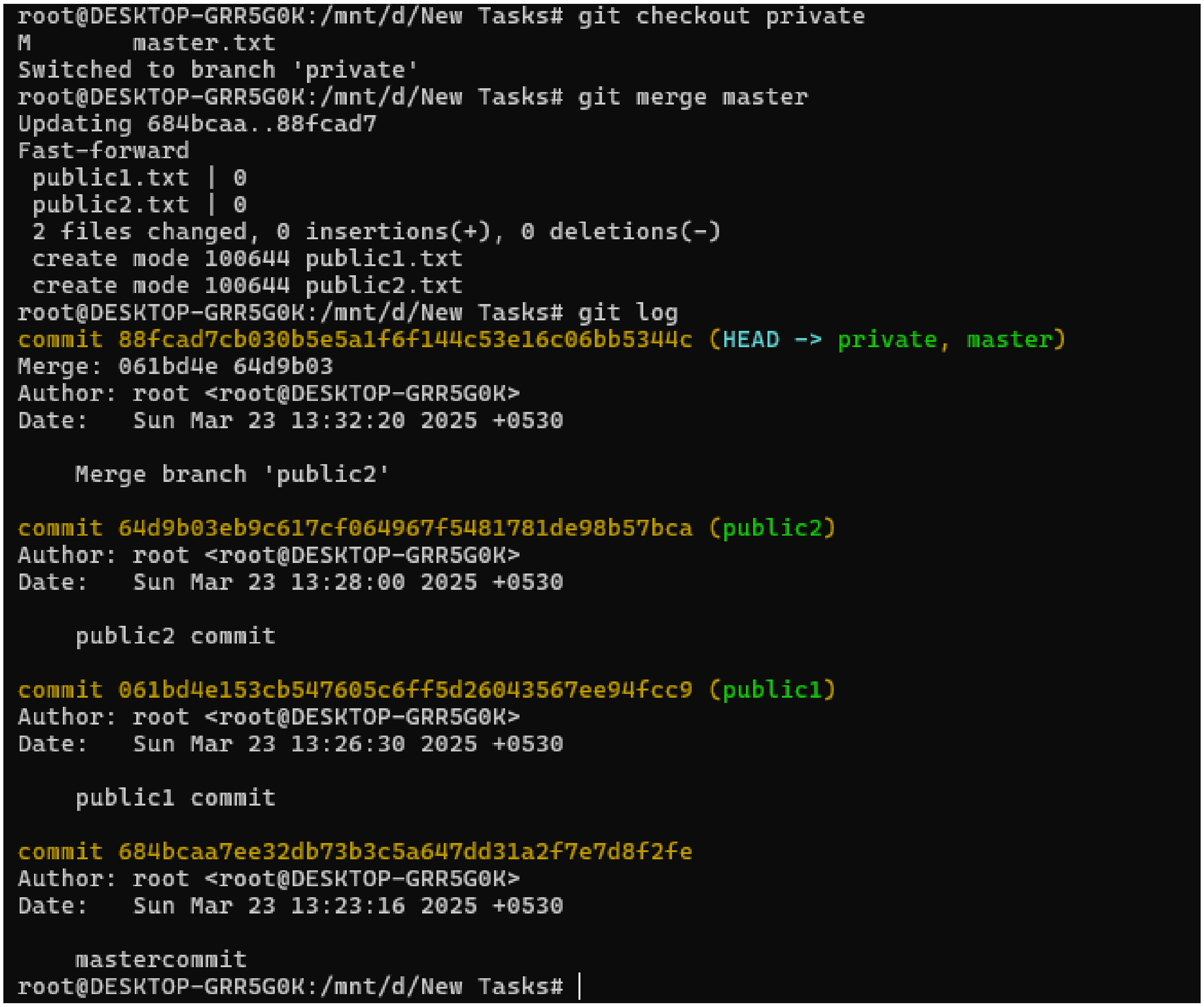
8 Also update new master code on master

git checkout master

git merge private

9 Finally update all the code on the private branch

For this we will not checkout to private branch and merge the master branch



```
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git checkout private
M       master.txt
Switched to branch 'private'
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git merge master
Updating 684bcaa..88fcad7
Fast-forward
 public1.txt | 0
 public2.txt | 0
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 public1.txt
 create mode 100644 public2.txt
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# git log
commit 88fcad7cb030b5e5a1f6f144c53e16c06bb5344c (HEAD -> private, master)
Merge: 061bd4e 64d9b03
Author: root <root@DESKTOP-GRR5G0K>
Date:   Sun Mar 23 13:32:20 2025 +0530

    Merge branch 'public2'

commit 64d9b03eb9c617cf064967f5481781de98b57bca (public2)
Author: root <root@DESKTOP-GRR5G0K>
Date:   Sun Mar 23 13:28:00 2025 +0530

    public2 commit

commit 061bd4e153cb547605c6ff5d26043567ee94fcc9 (public1)
Author: root <root@DESKTOP-GRR5G0K>
Date:   Sun Mar 23 13:26:30 2025 +0530

    public1 commit

commit 684bcaa7ee32db73b3c5a647dd31a2f7e7d8f2fe
Author: root <root@DESKTOP-GRR5G0K>
Date:   Sun Mar 23 13:23:16 2025 +0530

    mastercommit
root@DESKTOP-GRR5G0K:/mnt/d/New Tasks# 
```

Q5. Tasks to Be Performed:

1. Create a Git Flow workflow architecture on Git

2. Create all the required branches

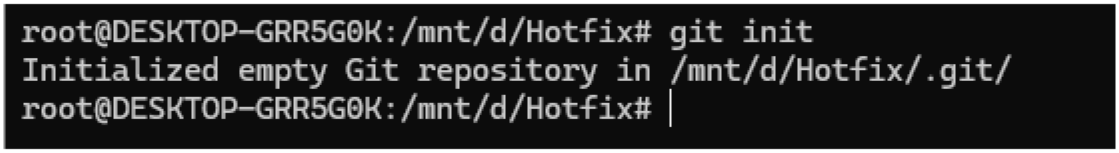
3. starting from e feature branch, push the branch to the master, following the architecture

4. Push an urgent.txt on master using hotfix

Git Flow is a branching model for Git that helps manage development, releases, and hotfixes systematically. The main branches are:

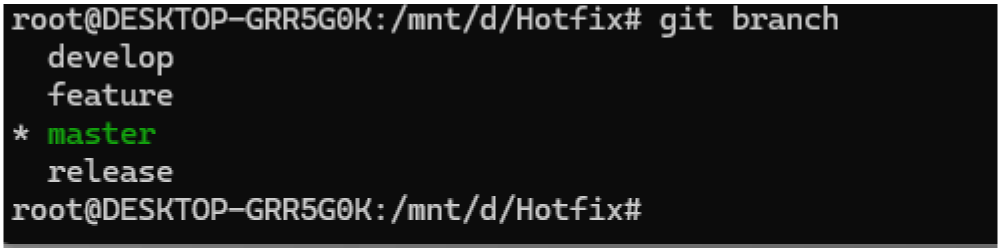
- **master** → The production branch, only updated through releases or hotfixes.
- **develop** → The main branch for ongoing development.
- **Feature branches (feature/*)** → For new features.
- **Release branches (release/*)** → For preparing stable releases.
- **Hotfix branches (hotfix/*)** → For urgent fixes directly applied to master.

First, initialize Git Flow in your repository:



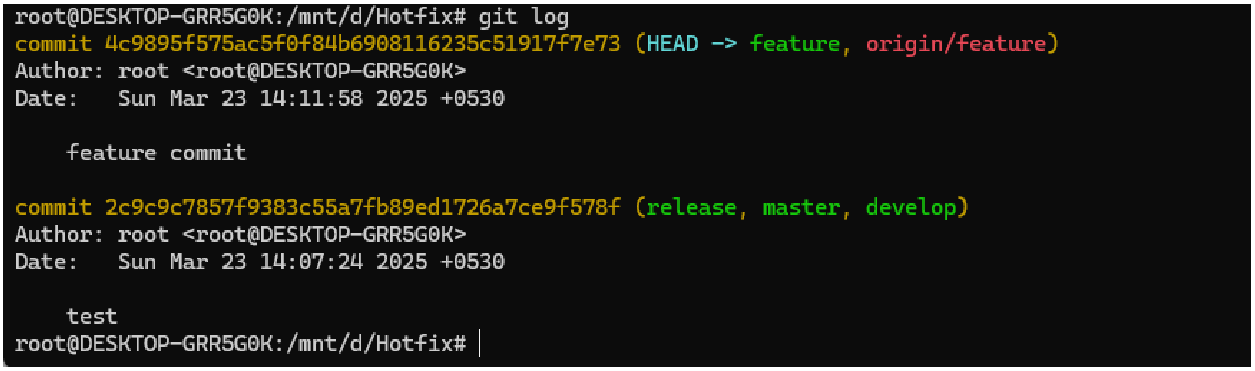
```
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix# git init
Initialized empty Git repository in /mnt/d/Hotfix/.git/
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix# |
```

Create Required Branches

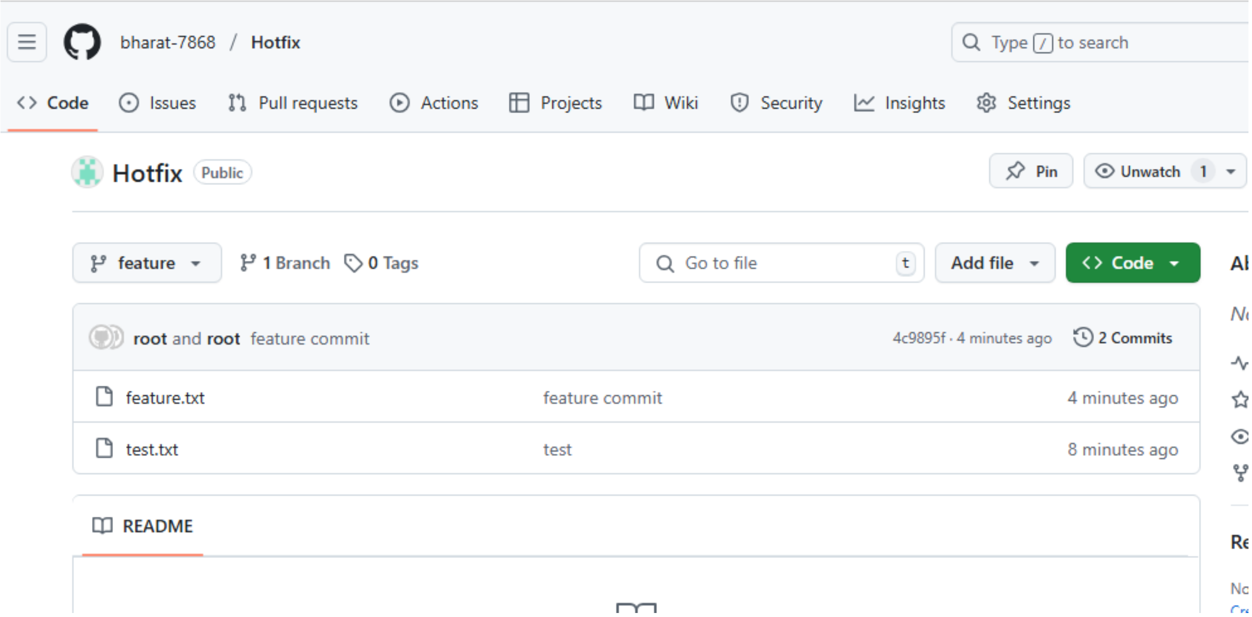


```
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix# git branch
  develop
  feature
* master
  release
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix#
```

Push a Feature Branch to Master (Following Git Flow)



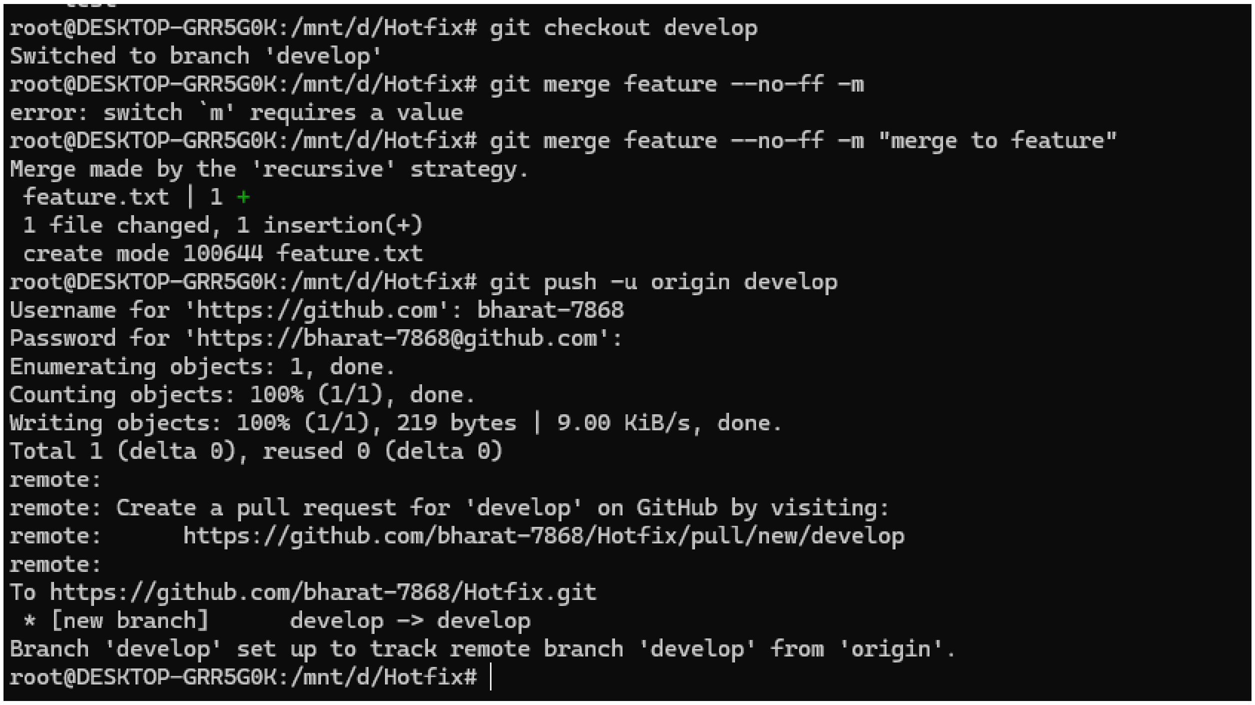
```
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix# git log
commit 4c9895f575ac5f0f84b6908116235c51917f7e73 (HEAD -> feature, origin/feature)
Author: root <root@DESKTOP-GRR5G0K>
Date:   Sun Mar 23 14:11:58 2025 +0530

    feature commit

commit 2c9c9c7857f9383c55a7fb89ed1726a7ce9f578f (release, master, develop)
Author: root <root@DESKTOP-GRR5G0K>
Date:   Sun Mar 23 14:07:24 2025 +0530

    test
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix# |
```



root and root feature commit 4c9895f · 4 minutes ago 2 Commits

feature.txt	feature commit	4 minutes ago
test.txt	test	8 minutes ago

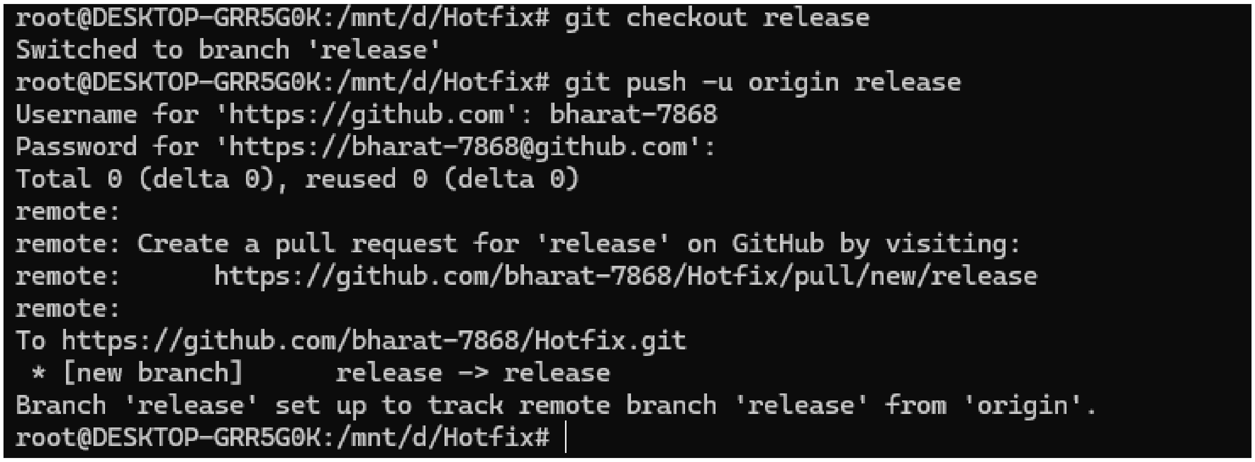
Merge feature branch to develop branch then push



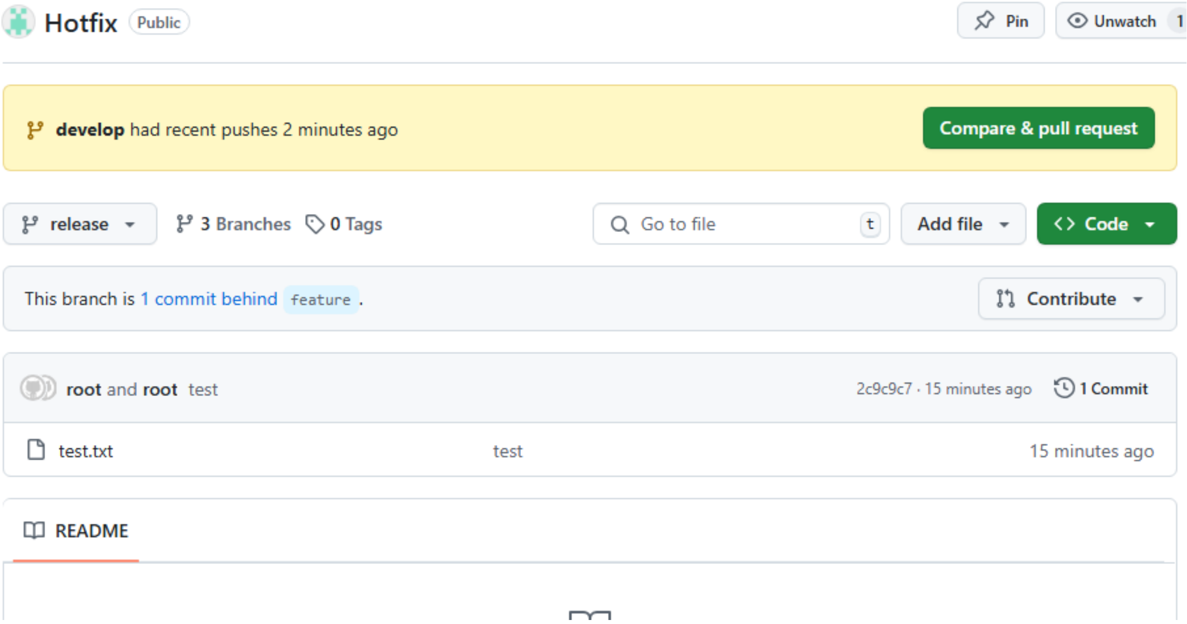
```
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix# git checkout develop
Switched to branch 'develop'
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix# git merge feature --no-ff -m
error: switch `m' requires a value
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix# git merge feature --no-ff -m "merge to feature"
Merge made by the 'recursive' strategy.
 feature.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 feature.txt
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix# git push -u origin develop
Username for 'https://github.com': bharat-7868
Password for 'https://bharat-7868@github.com':
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 219 bytes | 9.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:      https://github.com/bharat-7868/Hotfix/pull/new/develop
remote:
To https://github.com/bharat-7868/Hotfix.git
 * [new branch]      develop -> develop
Branch 'develop' set up to track remote branch 'develop' from 'origin'.
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix# 
```

Prepare a Release

Now we will switch to release branch then we will push to git hub repository



```
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix# git checkout release
Switched to branch 'release'
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix# git push -u origin release
Username for 'https://github.com': bharat-7868
Password for 'https://bharat-7868@github.com':
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'release' on GitHub by visiting:
remote:      https://github.com/bharat-7868/Hotfix/pull/new/release
remote:
To https://github.com/bharat-7868/Hotfix.git
 * [new branch]      release -> release
Branch 'release' set up to track remote branch 'release' from 'origin'.
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix#
```



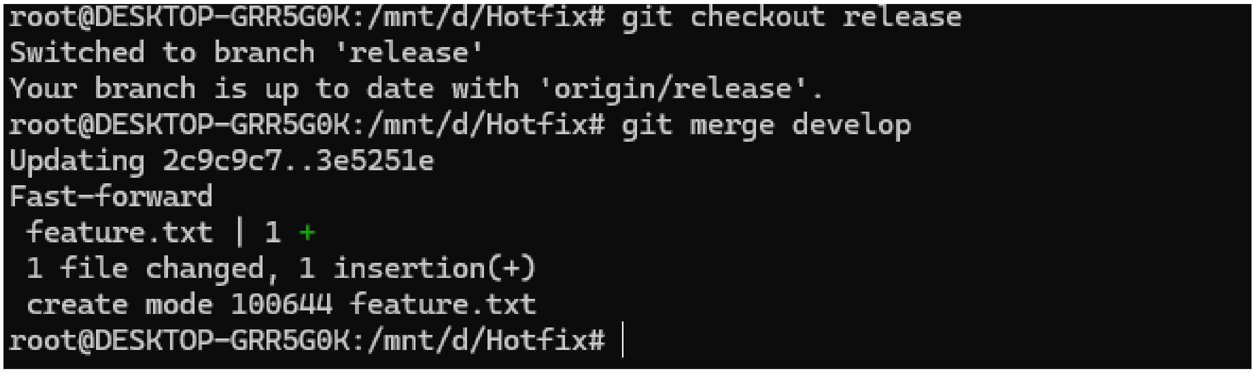
develop had recent pushes 2 minutes ago

This branch is 1 commit behind `feature` .

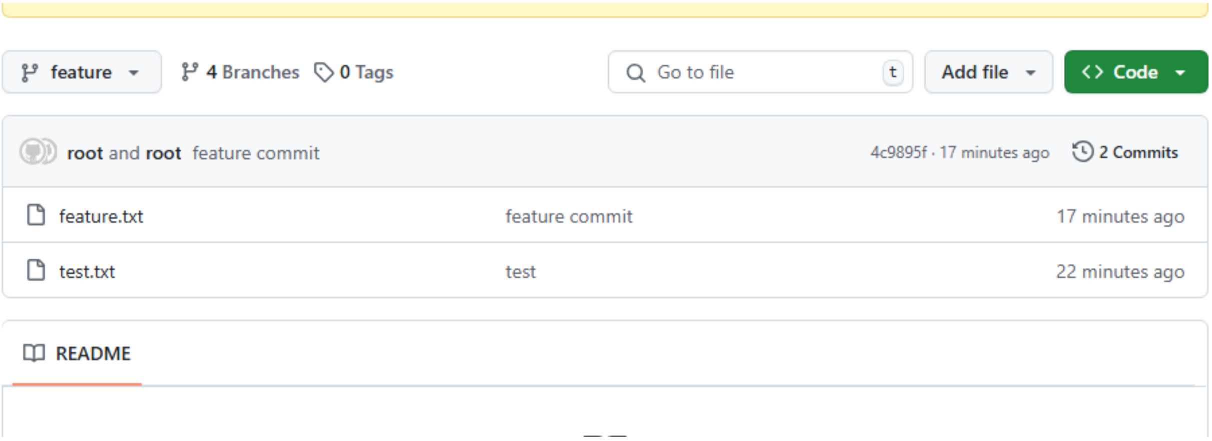
root and root test 2c9c9c7 · 15 minutes ago

test.txt test 15 minutes ago

Now we will merge develop branch to the release branch so that we can get the files in release branch as well.



```
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix# git checkout release
Switched to branch 'release'
Your branch is up to date with 'origin/release'.
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix# git merge develop
Updating 2c9c9c7..3e5251e
Fast-forward
 feature.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 feature.txt
root@DESKTOP-GRR5G0K:/mnt/d/Hotfix#
```



root and root feature commit 4c9895f · 17 minutes ago 2 Commits

feature.txt	feature commit	17 minutes ago
test.txt	test	22 minutes ago

Hence we can see that merging is happening as shown in the above picture.

Thankyou.