1. **Fork below public repository to your account. Then clone the repository locally. Create 2 branches cloud_dev01. Then add cloud_dev02 to that repository locally and hello.txt , moto.txt and world.txt to one branch and add another 3 files to another branch. Then push cloud_dev02 branch to the Forked repository on the remote. https://github.com/ripienaar/free-for-dev. ANSWER:-**

Note:- that this method does not retain issues, pull requests, or other settings from the original repository.

A) GitHub will prevent you from forking within your own repository because you own the original repository, and forking within the same account is not allowed.

B) So, we need to perform some extra activity**.**

so we create new repo then ➔settings➔danger-zone➔archive.

C) Then we need to create another repo and then clone the archived repo to the remote terminal.

D) And then copy the archived repo to the newly created repo and perform repo on that.

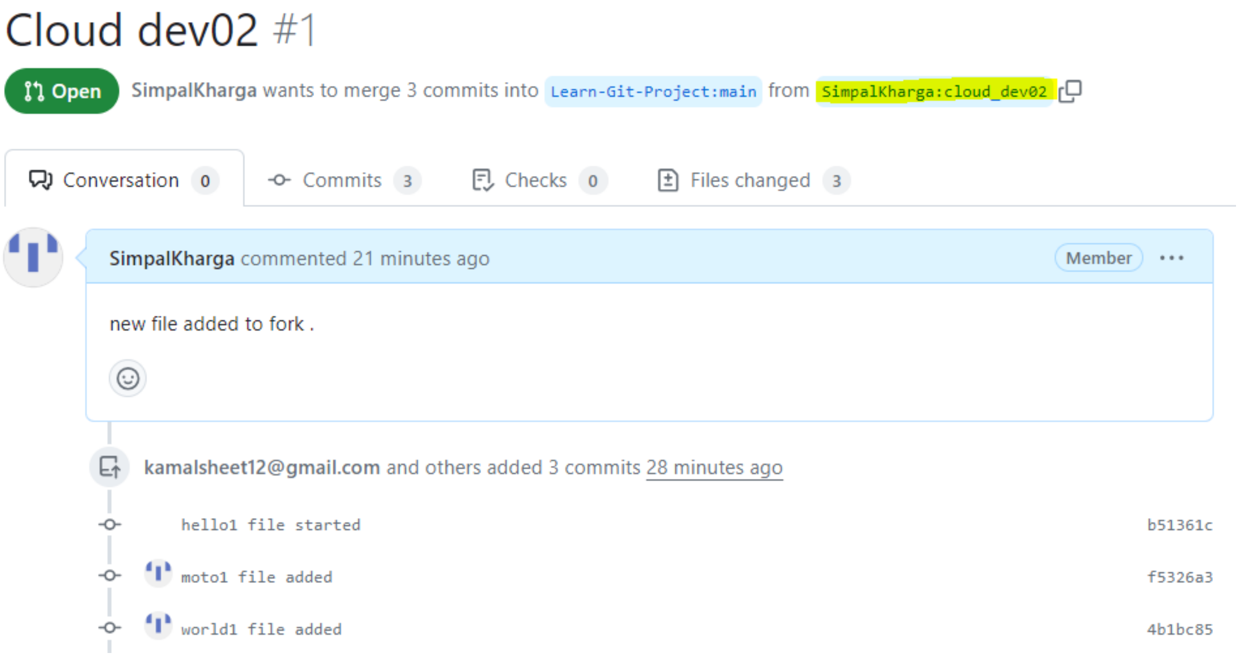
OR:-

You can directly create an organization under your Github account going under settings.

Then you can easily fork from there.

For cloning:-

use the SSH key to clone using :- git clone <ssh_key>



Cloud dev02 #1

SimpalKharga wants to merge 3 commits into Learn-Git-Project:main from SimpalKharga:cloud_dev02

SimpalKharga commented 21 minutes ago

new file added to fork .

kamalsheet12@gmail.com and others added 3 commits 28 minutes ago

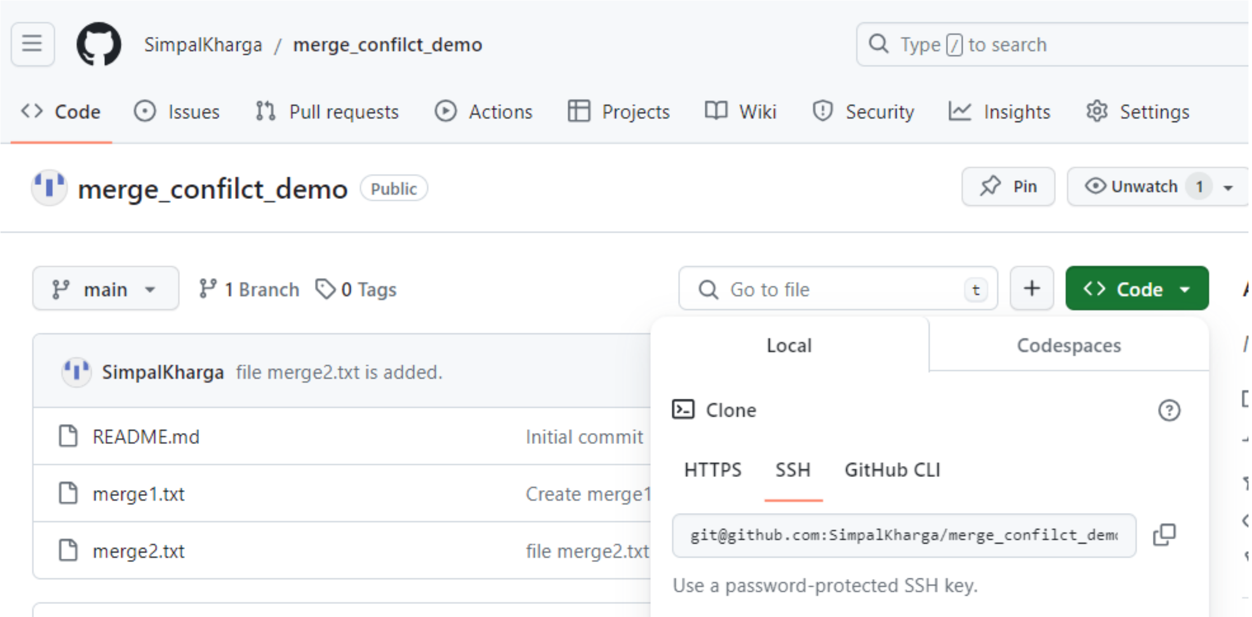
```
hello1 file started     b51361c
moto1 file added        f5326a3
world1 file added       4b1bc85
```

2. Create a remote repository merge_confilct_demo. Add some files in the remote master branch. Clone the repository locally. Create 2 branches locally named demo/merge and demo/conflicts. Add a merge.txt file in both branches with some data like "Hello CloudEthiX…." And "Fello CloudEthiX…." respectively and commit. Then merge demo/conflicts branch to demo/merge branch and solve the merge conflicts.

ANSWER :-

Create a remote repository merge_confilct_demo. Add some files in the remote master branch.

Created a repo on remote repository and using the add file sign on dashboard, we created two files "merge1.txt &merge2.txt" and commit the changes and also adeed the content inside it.



SimpalKharga file merge2.txt is added.

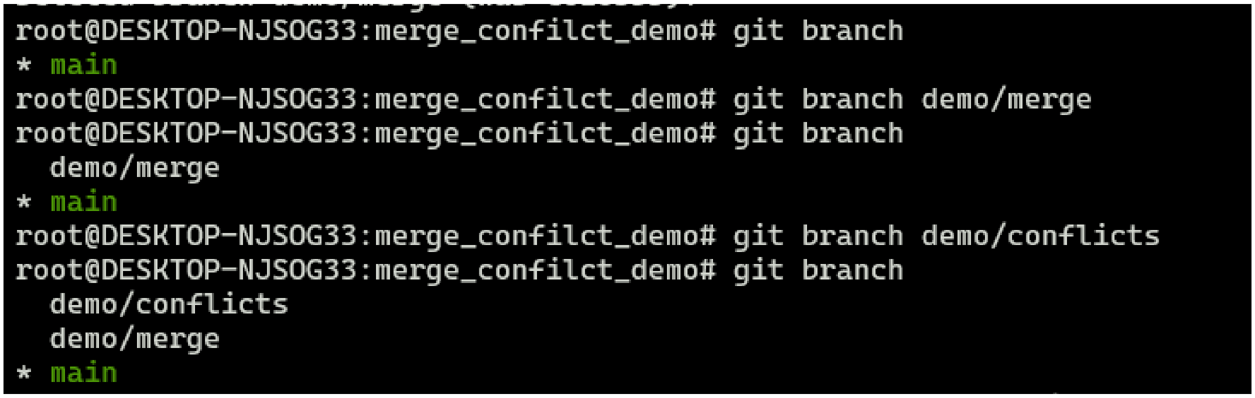
README.md	Initial commit
merge1.txt	Create merge1
merge2.txt	file merge2.txt

```
git@github.com:SimpalKharga/merge_confilct_demo
```

Use a password-protected SSH key.

Create 2 branches locally named demo/merge and demo/conflicts.

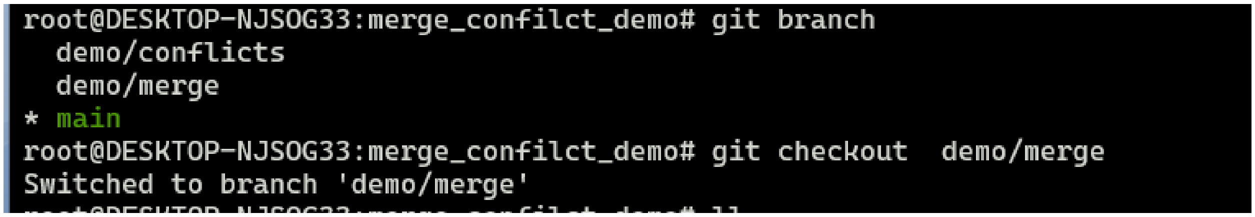
Make sure to create the branch under “main” only.
So. Being under main run below commands and create 2 branchesvas asked.



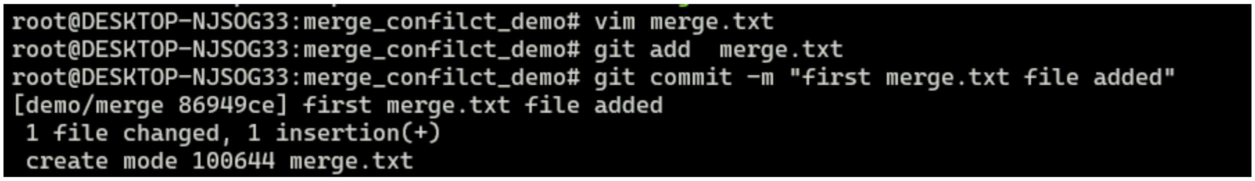
```
root@DESKTOP-NJSOG33:merge_confilct_demo# git branch
* main
root@DESKTOP-NJSOG33:merge_confilct_demo# git branch demo/merge
root@DESKTOP-NJSOG33:merge_confilct_demo# git branch
  demo/merge
* main
root@DESKTOP-NJSOG33:merge_confilct_demo# git branch demo/conflicts
root@DESKTOP-NJSOG33:merge_confilct_demo# git branch
  demo/conflicts
  demo/merge
* main
```

Add a merge.txt file in both branches with some data like "Hello CloudEthiX...." And "Fello CloudEthiX...." respectively

and commit.

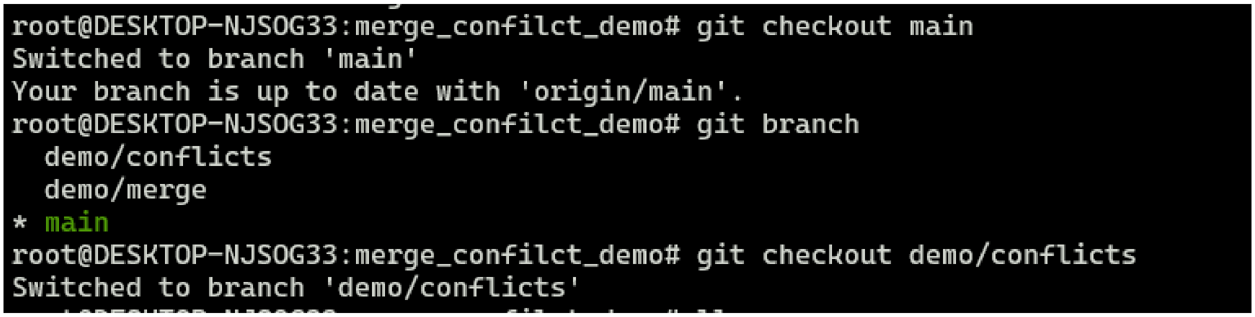


```
root@DESKTOP-NJSOG33:merge_confilct_demo# git branch
  demo/conflicts
  demo/merge
* main
root@DESKTOP-NJSOG33:merge_confilct_demo# git checkout  demo/merge
Switched to branch 'demo/merge'
root@DESKTOP-NJSOG33:merge_confilct_demo# ll
```



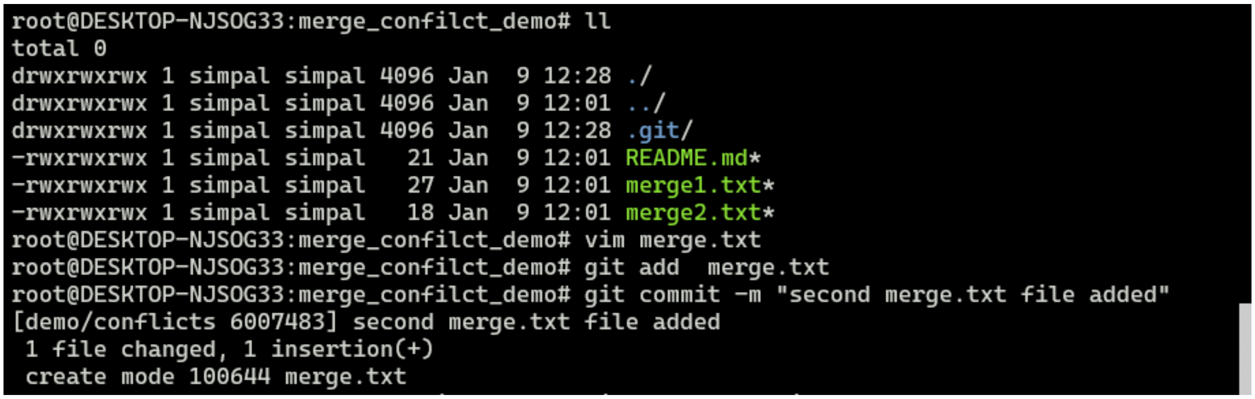
```
root@DESKTOP-NJSOG33:merge_confilct_demo# vim merge.txt
root@DESKTOP-NJSOG33:merge_confilct_demo# git add  merge.txt
root@DESKTOP-NJSOG33:merge_confilct_demo# git commit -m "first merge.txt file added"
[demo/merge 86949ce] first merge.txt file added
 1 file changed, 1 insertion(+)
 create mode 100644 merge.txt
```

Now go to second branch by switching to the main and then to the second branch.



```
root@DESKTOP-NJSOG33:merge_confilct_demo# git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
root@DESKTOP-NJSOG33:merge_confilct_demo# git branch
  demo/conflicts
  demo/merge
* main
root@DESKTOP-NJSOG33:merge_confilct_demo# git checkout demo/conflicts
Switched to branch 'demo/conflicts'
root@DESKTOP-NJSOG33:merge_confilct_demo# ll
```

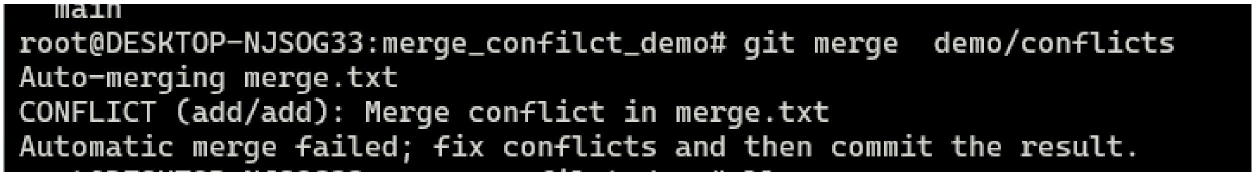
Note:- run “ll” to check the files present in it as the files of other sub branch shall not show in this branch .
And then create the other file in second branch “demo/conflicts”



```
root@DESKTOP-NJSOG33:merge_confilct_demo# ll
total 0
drwxrwxrwx 1 simpal simpal 4096 Jan  9 12:28 ./
drwxrwxrwx 1 simpal simpal 4096 Jan  9 12:01 ../
drwxrwxrwx 1 simpal simpal 4096 Jan  9 12:28 .git/
-rwxrwxrwx 1 simpal simpal   21 Jan  9 12:01 README.md*
-rwxrwxrwx 1 simpal simpal   27 Jan  9 12:01 merge1.txt*
-rwxrwxrwx 1 simpal simpal   18 Jan  9 12:01 merge2.txt*
root@DESKTOP-NJSOG33:merge_confilct_demo# vim merge.txt
root@DESKTOP-NJSOG33:merge_confilct_demo# git add  merge.txt
root@DESKTOP-NJSOG33:merge_confilct_demo# git commit -m "second merge.txt file added"
[demo/conflicts 6007483] second merge.txt file added
 1 file changed, 1 insertion(+)
 create mode 100644 merge.txt
```

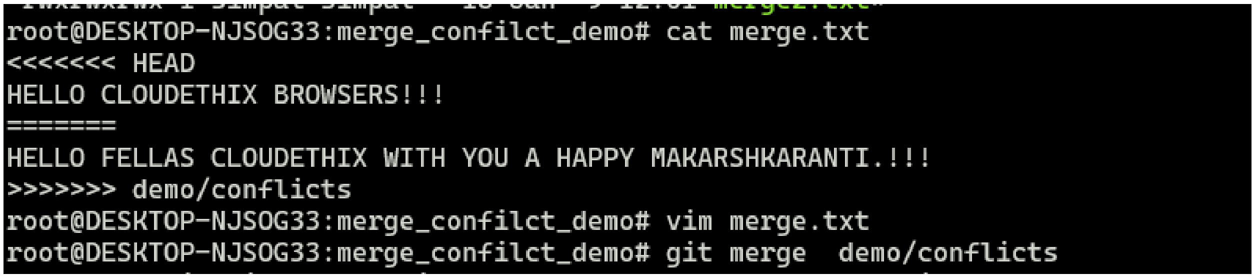
Then merge demo/conflicts branch to demo/merge branch and solve the merge conflicts.

A) Now go to the demo/conflicts branch using “checkout” command & run “git merge demo/conflicts” and it will display:



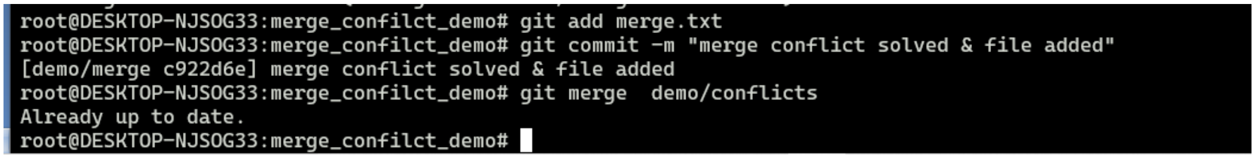
```
  main
root@DESKTOP-NJSOG33:merge_confilct_demo# git merge  demo/conflicts
Auto-merging merge.txt
CONFLICT (add/add): Merge conflict in merge.txt
Automatic merge failed; fix conflicts and then commit the result.
```

B) And now, ‘cat’ the file.



```
root@DESKTOP-NJSOG33:merge_confilct_demo# cat merge.txt
<<<<<<< HEAD
HELLO CLOUDETHIX BROWSERS!!!
=======
HELLO FELLAS CLOUDETHIX WITH YOU A HAPPY MAKARSHKARANTI.!!!
>>>>>>> demo/conflicts
root@DESKTOP-NJSOG33:merge_confilct_demo# vim merge.txt
root@DESKTOP-NJSOG33:merge_confilct_demo# git merge  demo/conflicts
```

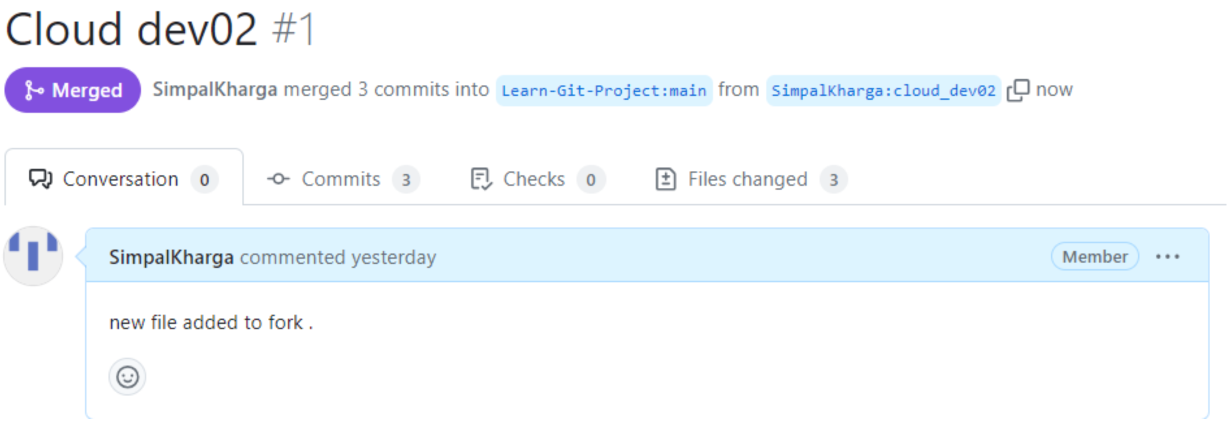
C) Now vim this file and delete the content, you don’t want and save.
And then add and commit.
Then run the “merge” command and check.



```
root@DESKTOP-NJSOG33:merge_confilct_demo# git add merge.txt
root@DESKTOP-NJSOG33:merge_confilct_demo# git commit -m "merge conflict solved & file added"
[demo/merge c922d6e] merge conflict solved & file added
root@DESKTOP-NJSOG33:merge_confilct_demo# git merge  demo/conflicts
Already up to date.
root@DESKTOP-NJSOG33:merge_confilct_demo# █
```

==

3. Create PR to merge the cloud_dev02 branch to Forked repository. Approve the PR and merge cloud_dev02 branch to master branch remotely.

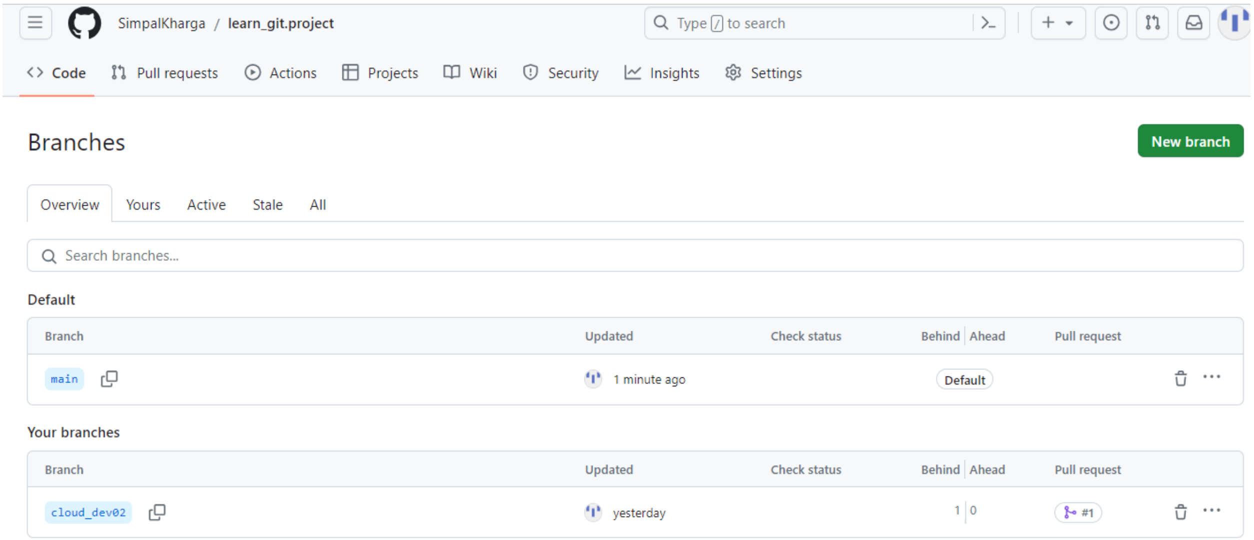


Cloud dev02 #1

Merged SimpalKharga merged 3 commits into `Learn-Git-Project:main` from `SimpalKharga:cloud_dev02` now

SimpalKharga commented yesterday — Member

new file added to fork .



Branches

Default

Branch	Updated	Check status	Behind | Ahead	Pull request
main	1 minute ago		Default	

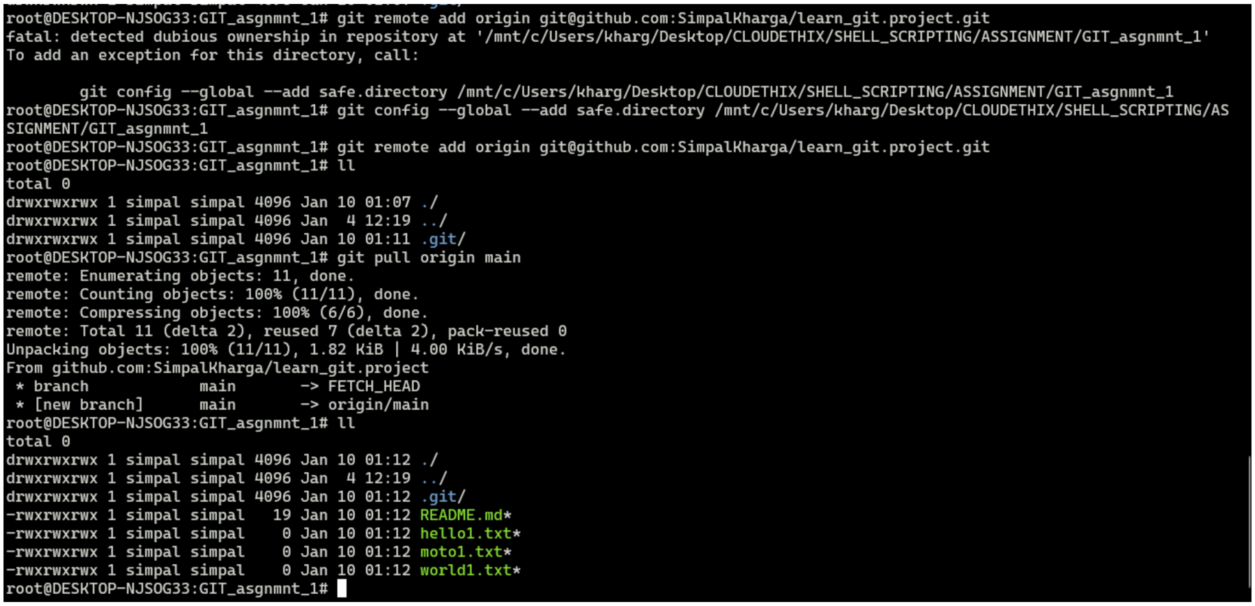
Your branches

Branch	Updated	Check status	Behind | Ahead	Pull request
cloud_dev02	yesterday		1 | 0	#1

4. Fetch latest changes from master branch of remote forked repository to local master branch. Then merge forked remote master branch with local master branch. Check If all the latest files are there. Then checkout to local cloud_dev03 branch locally. Then Pull latest changes from the remote master branch.

Solution:-

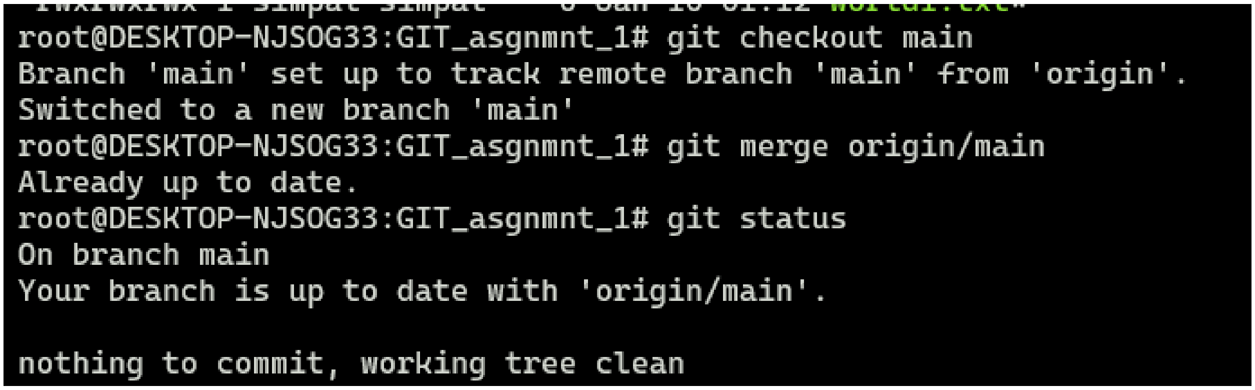
Fetch latest changes from master branch of remote forked repository to local master branch



```
root@DESKTOP-NJSOG33:GIT_asgnmnt_1# git remote add origin git@github.com:SimpalKharga/learn_git.project.git
fatal: detected dubious ownership in repository at '/mnt/c/Users/kharg/Desktop/CLOUDETHIX/SHELL_SCRIPTING/ASSIGNMENT/GIT_asgnmnt_1'
To add an exception for this directory, call:

        git config --global --add safe.directory /mnt/c/Users/kharg/Desktop/CLOUDETHIX/SHELL_SCRIPTING/ASSIGNMENT/GIT_asgnmnt_1
root@DESKTOP-NJSOG33:GIT_asgnmnt_1# git config --global --add safe.directory /mnt/c/Users/kharg/Desktop/CLOUDETHIX/SHELL_SCRIPTING/AS
SIGNMENT/GIT_asgnmnt_1
root@DESKTOP-NJSOG33:GIT_asgnmnt_1# git remote add origin git@github.com:SimpalKharga/learn_git.project.git
root@DESKTOP-NJSOG33:GIT_asgnmnt_1# ll
total 0
drwxrwxrwx 1 simpal simpal 4096 Jan 10 01:07 ./
drwxrwxrwx 1 simpal simpal 4096 Jan  4 12:19 ../
drwxrwxrwx 1 simpal simpal 4096 Jan 10 01:11 .git/
root@DESKTOP-NJSOG33:GIT_asgnmnt_1# git pull origin main
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 11 (delta 2), reused 7 (delta 2), pack-reused 0
Unpacking objects: 100% (11/11), 1.82 KiB | 4.00 KiB/s, done.
From github.com:SimpalKharga/learn_git.project
 * branch            main       -> FETCH_HEAD
 * [new branch]      main       -> origin/main
root@DESKTOP-NJSOG33:GIT_asgnmnt_1# ll
total 0
drwxrwxrwx 1 simpal simpal 4096 Jan 10 01:12 ./
drwxrwxrwx 1 simpal simpal 4096 Jan  4 12:19 ../
drwxrwxrwx 1 simpal simpal 4096 Jan 10 01:12 .git/
-rwxrwxrwx 1 simpal simpal   19 Jan 10 01:12 README.md*
-rwxrwxrwx 1 simpal simpal    0 Jan 10 01:12 hello1.txt*
-rwxrwxrwx 1 simpal simpal    0 Jan 10 01:12 moto1.txt*
-rwxrwxrwx 1 simpal simpal    0 Jan 10 01:12 world1.txt*
root@DESKTOP-NJSOG33:GIT_asgnmnt_1#
```

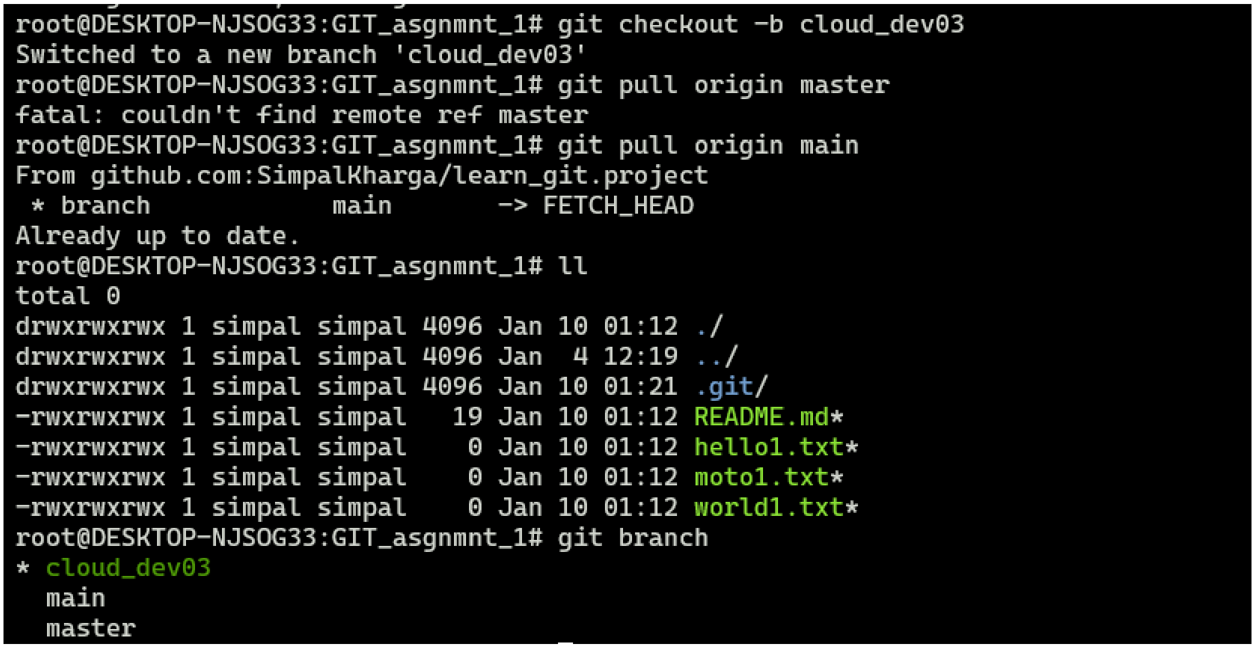
Then merge forked remote master branch with local master branch.



```
root@DESKTOP-NJSOG33:GIT_asgnmnt_1# git checkout main
Branch 'main' set up to track remote branch 'main' from 'origin'.
Switched to a new branch 'main'
root@DESKTOP-NJSOG33:GIT_asgnmnt_1# git merge origin/main
Already up to date.
root@DESKTOP-NJSOG33:GIT_asgnmnt_1# git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Then checkout to local cloud_dev03 branch locally. Then Pull latest changes from the remote master branch.



```
root@DESKTOP-NJSOG33:GIT_asgnmnt_1# git checkout -b cloud_dev03
Switched to a new branch 'cloud_dev03'
root@DESKTOP-NJSOG33:GIT_asgnmnt_1# git pull origin master
fatal: couldn't find remote ref master
root@DESKTOP-NJSOG33:GIT_asgnmnt_1# git pull origin main
From github.com:SimpalKharga/learn_git.project
 * branch            main       -> FETCH_HEAD
Already up to date.
root@DESKTOP-NJSOG33:GIT_asgnmnt_1# ll
total 0
drwxrwxrwx 1 simpal simpal 4096 Jan 10 01:12 ./
drwxrwxrwx 1 simpal simpal 4096 Jan  4 12:19 ../
drwxrwxrwx 1 simpal simpal 4096 Jan 10 01:21 .git/
-rwxrwxrwx 1 simpal simpal   19 Jan 10 01:12 README.md*
-rwxrwxrwx 1 simpal simpal    0 Jan 10 01:12 hello1.txt*
-rwxrwxrwx 1 simpal simpal    0 Jan 10 01:12 moto1.txt*
-rwxrwxrwx 1 simpal simpal    0 Jan 10 01:12 world1.txt*
root@DESKTOP-NJSOG33:GIT_asgnmnt_1# git branch
* cloud_dev03
  main
  master
```

===

5. Checkout to the cloud_dev01 branch locally and rebase the remote master branch of the forked repository.

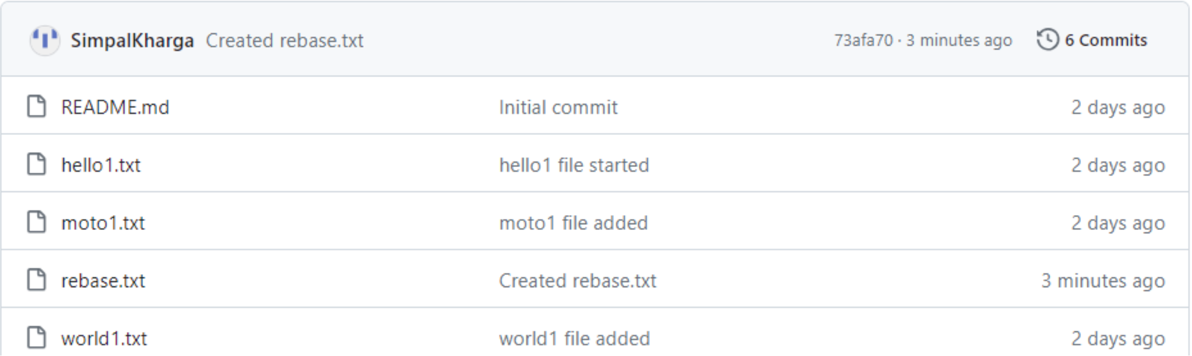
Solution:-

First clone the remote repo.
And go to the Branch cloud_dev01.

NOTE:-
Rebase basically means, let's take an example:
There is a repo on central repository and we create one sub branch under main i.e. Release and on that tow developers are working one finished his work little early and merge his locally created branch to the main branch and in that case the other didn't have the access to the latest changes.

Then the other developer will run the ‘rebase’ command to fetch the newly edited changes to its local repository.

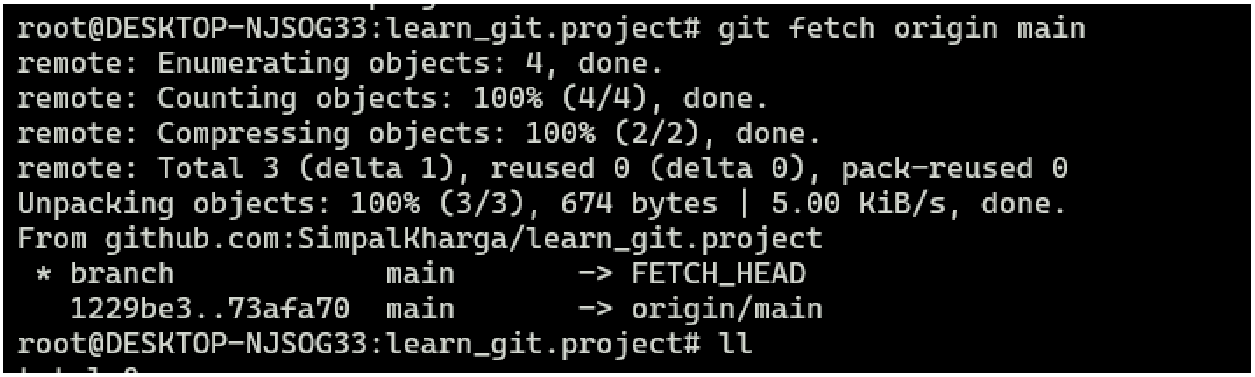
A) Let’s first create a file on main branch on remote repo on git hub.



SimpalKharga Created rebase.txt — 73afa70 · 3 minutes ago — 6 Commits

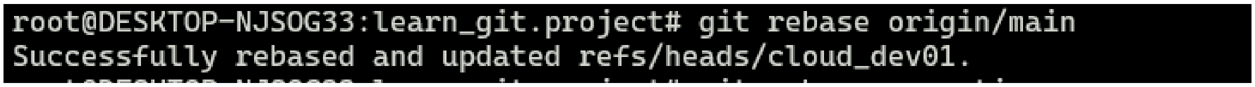
README.md	Initial commit	2 days ago
hello1.txt	hello1 file started	2 days ago
moto1.txt	moto1 file added	2 days ago
rebase.txt	Created rebase.txt	3 minutes ago
world1.txt	world1 file added	2 days ago

B) Then again go to the local repo and run



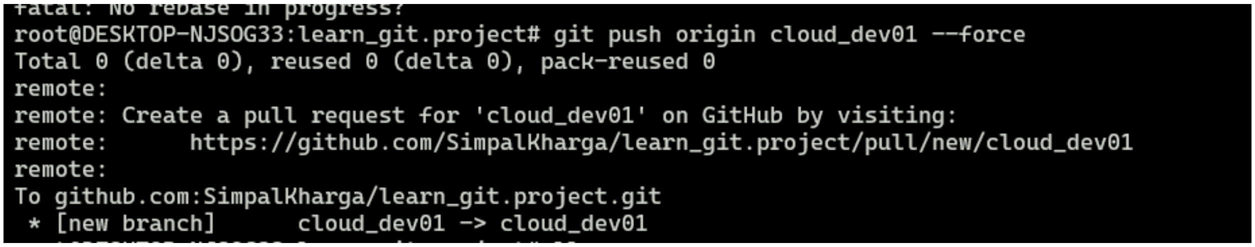
```
root@DESKTOP-NJSOG33:learn_git.project# git fetch origin main
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 674 bytes | 5.00 KiB/s, done.
From github.com:SimpalKharga/learn_git.project
 * branch            main       -> FETCH_HEAD
   1229be3..73afa70  main       -> origin/main
root@DESKTOP-NJSOG33:learn_git.project# ll
```

C)

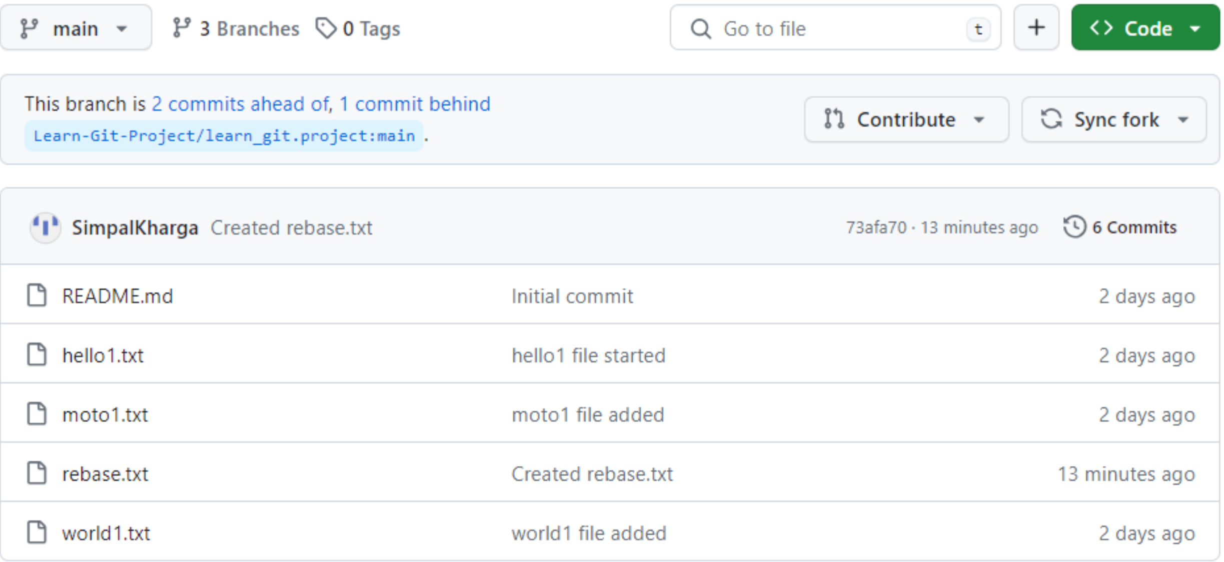


```
root@DESKTOP-NJSOG33:learn_git.project# git rebase origin/main
Successfully rebased and updated refs/heads/cloud_dev01.
```

D)



```
fatal: No rebase in progress?
root@DESKTOP-NJSOG33:learn_git.project# git push origin cloud_dev01 --force
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'cloud_dev01' on GitHub by visiting:
remote:      https://github.com/SimpalKharga/learn_git.project/pull/new/cloud_dev01
remote:
To github.com:SimpalKharga/learn_git.project.git
 * [new branch]      cloud_dev01 -> cloud_dev01
```

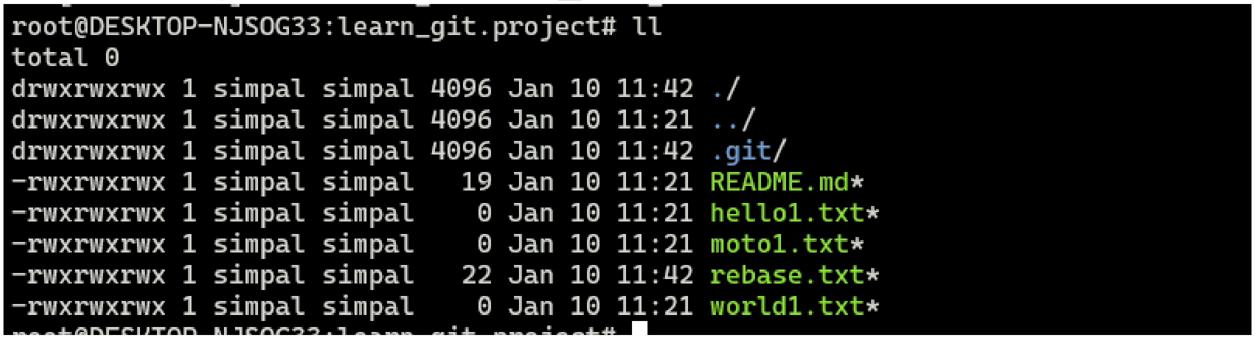


This branch is 2 commits ahead of, 1 commit behind Learn-Git-Project/learn_git.project:main .

SimpalKharga Created rebase.txt 73afa70 · 13 minutes ago

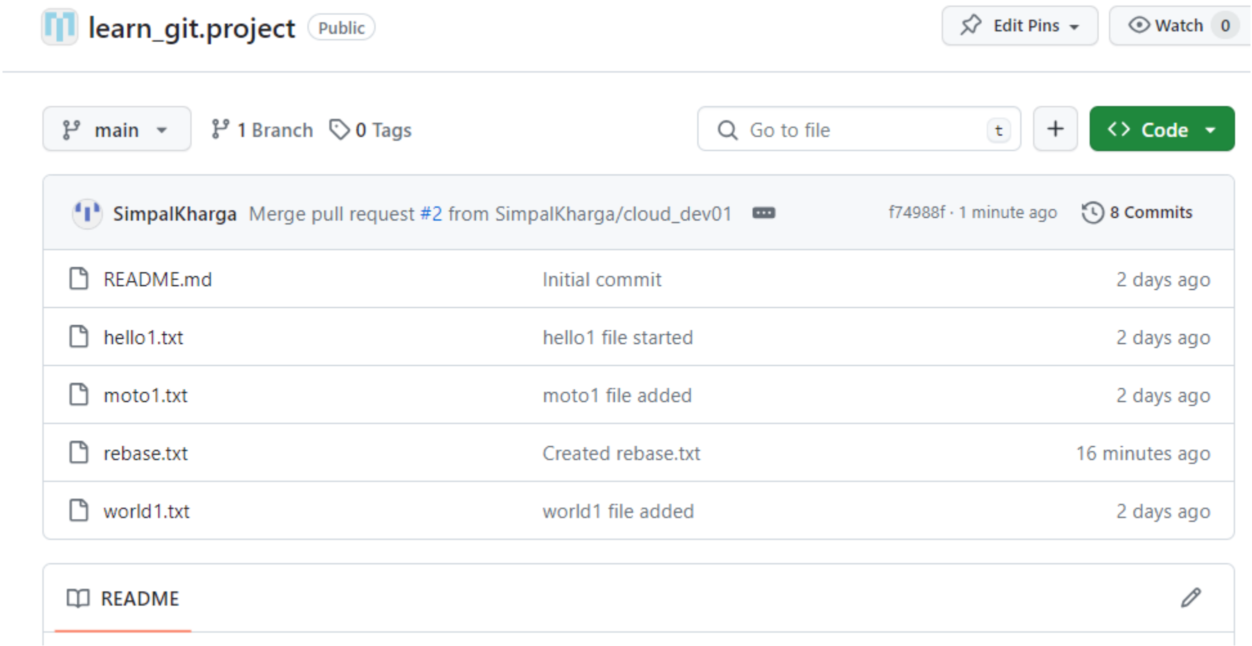
README.md	Initial commit	2 days ago
hello1.txt	hello1 file started	2 days ago
moto1.txt	moto1 file added	2 days ago
rebase.txt	Created rebase.txt	13 minutes ago
world1.txt	world1 file added	2 days ago

E)The newly edited 'rebase.txt' has been successfully bought to the local repo's branch cloud_dev01.



```
root@DESKTOP-NJSOG33:learn_git.project# ll
total 0
drwxrwxrwx 1 simpal simpal 4096 Jan 10 11:42 ./
drwxrwxrwx 1 simpal simpal 4096 Jan 10 11:21 ../
drwxrwxrwx 1 simpal simpal 4096 Jan 10 11:42 .git/
-rwxrwxrwx 1 simpal simpal   19 Jan 10 11:21 README.md*
-rwxrwxrwx 1 simpal simpal    0 Jan 10 11:21 hello1.txt*
-rwxrwxrwx 1 simpal simpal    0 Jan 10 11:21 moto1.txt*
-rwxrwxrwx 1 simpal simpal   22 Jan 10 11:42 rebase.txt*
-rwxrwxrwx 1 simpal simpal    0 Jan 10 11:21 world1.txt*
root@DESKTOP-NJSOG33:learn_git.project#
```

And also have merged the new edits in dev_clod01 to the main branch of original repo.



learn_git.project

SimpalKharga Merge pull request #2 from SimpalKharga/cloud_dev01 f74988f · 1 minute ago 8 Commits

README.md	Initial commit	2 days ago
hello1.txt	hello1 file started	2 days ago
moto1.txt	moto1 file added	2 days ago
rebase.txt	Created rebase.txt	16 minutes ago
world1.txt	world1 file added	2 days ago

6. Create one repository locally. Commit any 3 files to the local repository. And the revert back the commits.

Solution:-

A) Created ‘revert_repo_project’ on local repo

B) Run ‘git init’ .

C) Make three files under the main branch. (file1.txt and file2.txt) here I have the main branch named as master so I changed the name first using < git branch -m master main>.

D) Then add <git add .> and git commit < git commit -m "Initial commit with three files">

E) Revert the last commit.

And once the revert is done, it means it came down to staging area Once again and you can commit again.

The process below is to how to revert commit

```
   drwxrwxrwx 1 simpal simpal 4096 Jan 10 12:05 ../
drwxrwxrwx 1 simpal simpal 4096 Jan 10 12:10 .git/
-rwxrwxrwx 1 simpal simpal   19 Jan 10 12:15 file1.txt*
-rwxrwxrwx 1 simpal simpal   19 Jan 10 12:16 file2.txt*
simpal@DESKTOP-NJSOG33:revert_repo_project$ git add .
simpal@DESKTOP-NJSOG33:revert_repo_project$ git commit -m "Initial commit with three
files"
[main (root-commit) 5a42c6e] Initial commit with three files
 2 files changed, 2 insertions(+)
 create mode 100644 file1.txt
 create mode 100644 file2.txt
simpal@DESKTOP-NJSOG33:revert_repo_project$ git log --oneline
5a42c6e (HEAD -> main) Initial commit with three files
simpal@DESKTOP-NJSOG33:revert_repo_project$ git revert 5a42c6e
[main d12529d] Revert "Initial commit with three files"
 2 files changed, 2 deletions(-)
 delete mode 100644 file1.txt
 delete mode 100644 file2.txt
simpal@DESKTOP-NJSOG33:revert_repo_project$  git revert --no-edit 5a42c6e
On branch main
nothing to commit, working tree clean
simpal@DESKTOP-NJSOG33:revert_repo_project$ git log
commit d12529dedf5322c2dc023edc44b1efec538f2847 (HEAD -> main)
Author: Simpal kharga <kamalsheet12@gmail.com>
Date:   Wed Jan 10 12:23:32 2024 +0530

    Revert "Initial commit with three files"

    This reverts commit 5a42c6e1a112746a03517eb5971f772004a839ee.

commit 5a42c6e1a112746a03517eb5971f772004a839ee
Author: Simpal kharga <kamalsheet12@gmail.com>
Date:   Wed Jan 10 12:16:57 2024 +0530

    Initial commit with three files
```

7. Create one local repository. commit some files to the local master branch. Checkout to master_02 branch. Create 2 more files. Add those to the staging area. Create a stash area and list those files. Then list and delete the stash.

SOLUTION:-

```
root@DESKTOP-NJSOG33:ASSIGNMENT# mkdir my_local_repo
root@DESKTOP-NJSOG33:ASSIGNMENT# cd my_local_repo/
root@DESKTOP-NJSOG33:my_local_repo# git init


root@DESKTOP-NJSOG33:my_local_repo# touch data01 data02
root@DESKTOP-NJSOG33:my_local_repo# ll

root@DESKTOP-NJSOG33:my_local_repo# git add .
root@DESKTOP-NJSOG33:my_local_repo# git commit -m "adding files "
[master (root-commit) 47c6f17] adding files

root@DESKTOP-NJSOG33:my_local_repo# git checkout -b master_02branch
Switched to a new branch 'master_02branch'
root@DESKTOP-NJSOG33:my_local_repo# git branch
  master
* master_02branch

root@DESKTOP-NJSOG33:my_local_repo# touch data03 data04
root@DESKTOP-NJSOG33:my_local_repo# git add .
root@DESKTOP-NJSOG33:my_local_repo# ll
total 0
drwxrwxrwx 1 simpal simpal 4096 Jan 12 15:07 ./
drwxrwxrwx 1 simpal simpal 4096 Jan 12 14:56 ../
drwxrwxrwx 1 simpal simpal 4096 Jan 12 15:07 .git/
-rwxrwxrwx 1 simpal simpal    0 Jan 12 14:59 data01*
-rwxrwxrwx 1 simpal simpal    0 Jan 12 14:59 data02*
-rwxrwxrwx 1 simpal simpal    0 Jan 12 15:07 data03*
-rwxrwxrwx 1 simpal simpal    0 Jan 12 15:07 data04*
root@DESKTOP-NJSOG33:my_local_repo# git stash list
```

```
root@DESKTOP-NJSOG33:my_local_repo# git status
On branch master_02branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   data03
        new file:   data04

root@DESKTOP-NJSOG33:my_local_repo# git stash
Saved working directory and index state WIP on master_02branch: de5574f
Revert "adding 2 new files to o2branch"
root@DESKTOP-NJSOG33:my_local_repo# git stash list
stash@{0}: WIP on master_02branch: de5574f Revert "adding 2 new files to
o2branch"
root@DESKTOP-NJSOG33:my_local_repo# git stash drop
Dropped refs/stash@{0} (baa5e6ecdd57bb2101c307278552854aa9c35c6a)
root@DESKTOP-NJSOG33:my_local_repo#
```

Q8). Create a local repository, initialise it and create 3 branches: develop, hotfix and release. Checkout to hotfix branch and commit cloudethix.txt and once committed amend the commit message.

Solutions:-

```
root@DESKTOP-NJSOG33:ASSIGNMENT# mkdir my_local_repo
root@DESKTOP-NJSOG33:ASSIGNMENT# cd my_local_repo/
root@DESKTOP-NJSOG33:my_local_repo# git init
```

```
root@DESKTOP-NJSOG33:my_local_repo# git checkout -b develop
Switched to a new branch 'develop'
root@DESKTOP-NJSOG33:my_local_repo# git switch master
Switched to branch 'master'
root@DESKTOP-NJSOG33:my_local_repo# git branch hotfix
```

```
root@DESKTOP-NJSOG33:my_local_repo# git branch release
root@DESKTOP-NJSOG33:my_local_repo# git switch hotfix
Switched to branch 'hotfix'

root@DESKTOP-NJSOG33:my_local_repo# git branch
  develop
* hotfix
  master
  release
root@DESKTOP-NJSOG33:my_local_repo# vim cloudethix.txt
root@DESKTOP-NJSOG33:my_local_repo# git add cloudethix.txt
root@DESKTOP-NJSOG33:my_local_repo# git commit -m "this needs to be
ammend"
[hotfix c6603c5] this needs to be ammend
 1 file changed, 1 insertion(+)
 create mode 100644 cloudethix.txt

root@DESKTOP-NJSOG33:my_local_repo# git commit --amend -m "the file has
been ammended"
[hotfix caa87d3] the file has been ammended
 Date: Fri Jan 12 19:41:54 2024 +0530
 1 file changed, 1 insertion(+)
 create mode 100644 cloudethix.txt
root@DESKTOP-NJSOG33:my_local_repo#

=============================================================
```