**GIT demo**

**Importing a new project**

$ git init

Initialized empty Git repository in .git/

$ git add .

$ git commit

**Making changes**

Modify some files, then add their updated contents to the index:

$ git add file1 file2 file3

$ git diff --cached

$ git status

# On branch master

# Changes to be committed:

# (use "git reset HEAD <file>..." to unstage)

#

# modified: file1

# modified: file2

# modified: file3

#

$ git commit

$ git commit –a

**Viewing project history**

$ git log

$ git log -p

$ git log --stat –summary

**Git Branch and Merge:**

* **Merge**

D:\demo\_VC>cat intro.txt

git init - Local repository

\* create files

\* modify files

\* stage files

\* commit

\* .git folder is created - snapshots are present

D:\demo\_VC>git add .

D:\demo\_VC>git commit -m "file1 added"

[master b95a620] file1 added

1 file changed, 1 insertion(+)

create mode 100644 file1.txt

D:\demo\_VC>cat file1.txt

Helooooooooooo

// In testing branch I create a file testcase.txt

D:\demo\_VC>git checkout testing

Switched to branch 'testing'

D:\demo\_VC>cat intro.txt

git init - Local repository

\* create files

\* modify files

\* stage files

\* commit

\* .git folder is created - snapshots are present

D:\demo\_VC>cat file1.txt

cat: file1.txt: No such file or directory //Testing branch did not know about file1.txt

D:\demo\_VC>dir

Volume in drive D has no label.

Volume Serial Number is 44F8-5D6A

Directory of D:\demo\_VC

12/15/2014 12:27 PM <DIR> .

12/15/2014 12:27 PM <DIR> ..

12/15/2014 11:57 AM 374 cmds.txt

12/15/2014 11:57 AM 141 intro.txt

2 File(s) 515 bytes

2 Dir(s) 69,634,740,224 bytes free

D:\demo\_VC>touch testcase.txt

D:\demo\_VC>echo TC\_01 To test username>testcase.txt

D:\demo\_VC>git add .

D:\demo\_VC>git commit -m "test cases"

[testing 95a866a] test cases

1 file changed, 1 insertion(+)

create mode 100644 testcase.txt

//In master branch file testcase.txt is not visible

D:\demo\_VC>git checkout master

Switched to branch 'master'

D:\demo\_VC>echo testcase.txt

testcase.txt

D:\demo\_VC>cat testcase.txt

cat: testcase.txt: No such file or directory

D:\demo\_VC>dir

Volume in drive D has no label.

Volume Serial Number is 44F8-5D6A

Directory of D:\demo\_VC

12/15/2014 12:31 PM <DIR> .

12/15/2014 12:31 PM <DIR> ..

12/15/2014 11:57 AM 374 cmds.txt

12/15/2014 12:31 PM 14 file1.txt

12/15/2014 11:57 AM 141 intro.txt

3 File(s) 529 bytes

2 Dir(s) 69,634,723,840 bytes free

D:\demo\_VC>git branch

development

\* master

testing

D:\demo\_VC>git merge testing

Merge made by the 'recursive' strategy.

testcase.txt | 1 +

1 file changed, 1 insertion(+)

create mode 100644 testcase.txt

D:\demo\_VC>dir

Volume in drive D has no label.

Volume Serial Number is 44F8-5D6A

Directory of D:\demo\_VC

12/15/2014 12:33 PM <DIR> .

12/15/2014 12:33 PM <DIR> ..

12/15/2014 11:57 AM 374 cmds.txt

12/15/2014 12:31 PM 14 file1.txt

12/15/2014 11:57 AM 141 intro.txt

12/15/2014 12:33 PM 24 testcase.txt

4 File(s) 553 bytes

2 Dir(s) 69,634,723,840 bytes free

D:\demo\_VC>

**Create a new Branch:**

D:\dir1>git branch

\* master

D:\dir1>git branch testing

D:\dir1>git checkout testing

Switched to branch 'testing'

D:\dir1>echo "A cool new branch">page2.txt

D:\dir1>git commit -a -m "new feature"

[testing 95b0db6] new feature

1 file changed, 1 insertion(+), 1 deletion(-)

D:\dir1>git checkout master

Switched to branch 'master'

D:\dir1>cat page2.txt

Happy Weekend

D:\dir1>git checkout testing

Switched to branch 'testing'

D:\dir1>cat page2.txt

"A cool new branch"

D:\dir1>git checkout -b bugreport12

Switched to a new branch 'bugreport12'

D:\dir1>git checkout -b mybranch master~1

Switched to a new branch 'mybranch'

D:\dir1>git branch

bugreport12

master

\* mybranch

testing

D:\dir1>git checkout master

Switched to branch 'master'

D:\dir1>git branch

bugreport12

master

\* mybranch

testing

D:\dir1>git checkout master

Switched to branch 'master'

D:\dir1>git branch -m bugreport12 defectBranch

D:\dir1>git branch -d mybranch

Deleted branch mybranch (was cbbd0e4).

D:\dir1>git branch

defectBranch

\* master

testing

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Deleting and push on branches:**

D:\dir1>git branch

bugreport12

master

\* mybranch

testing

D:\dir1>git checkout master

Switched to branch 'master'

D:\dir1>git branch -m bugreport12 defectBranch

D:\dir1>git branch -d mybranch

Deleted branch mybranch (was cbbd0e4).

D:\dir1>git branch

defectBranch

\* master

Testing

D:\dir1>git push origin testing

fatal: 'origin' does not appear to be a git repository

fatal: Could not read from remote repository.

Please make sure you have the correct access rights

and the repository exists.

D:\dir1>git remote

D:\dir1>git push D:\dir2 testing

Counting objects: 17, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (12/12), done.

Writing objects: 100% (17/17), 1.63 KiB | 0 bytes/s, done.

Total 17 (delta 2), reused 0 (delta 0)

To D:\dir2

\* [new branch] testing -> testing

D:\dir1>git checkout testing

Switched to branch 'testing'

D:\dir1>echo "News for u">page1.txt

D:\dir1>cat page1.txt

"News for u"

D:\dir1>git commit -a -m "new feature in branch"

[testing a6b7b47] new feature in branch

1 file changed, 1 insertion(+), 4 deletions(-)

D:\dir1>git push

fatal: No configured push destination.

Either specify the URL from the command-line or configure a remote repository us

ing

git remote add <name> <url>

and then push using the remote name

git push <name>

D:\dir1>git push D:\dir2

fatal: The current branch testing has no upstream branch.

To push the current branch and set the remote as upstream, use

git push --set-upstream D:\dir2 testing

D:\dir1>git diff master testing

WARNING: terminal is not fully functional

diff --git a/page1.txt b/page1.txt

index 86fb49f..6e30bc3 100644

--- a/page1.txt

+++ b/page1.txt

@@ -1,4 +1 @@

-1 - India

-2 - USA

-3 - UK

-4 - Australia

+"News for u"

diff --git a/page2.txt b/page2.txt

index ff794d9..b55772d 100644

--- a/page2.txt

+++ b/page2.txt

@@ -1 +1 @@

-Happy Weekend

+"A cool new branch"

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Merging Branches demo:**

D:\gitDir>

D:\gitDir>git log

WARNING: terminal is not fully functional

commit d7cf814ac2fe33ba0a5c8061cb3600d1d51e0ebb

Author: Hema <Hema.g@igate.com>

Date: Mon Dec 8 12:07:49 2014 +0530

s Please enter the commit message for your changes. Lines starting

version 02:wq

:

commit 2af2ac14bb21309e33d7f1b6bbab08059d639274

Author: Hema <Hema.g@igate.com>

Date: Mon Dec 8 10:33:43 2014 +0530

W

D:\gitDir>git branch

\* master

D:\gitDir>git branch Testing

D:\gitDir>git branch

Testing

\* master

D:\gitDir>git checkout Testing

Switched to branch 'Testing'

D:\gitDir>git commit -a

Aborting commit due to empty commit message.

D:\gitDir>git commit -a

[Testing 70dc4df] New branch update

1 file changed, 1 insertion(+)

D:\gitDir>git log

WARNING: terminal is not fully functional

commit 70dc4df0593a8d0d8230ba711c2b014774f37873

Author: Hema <Hema.g@igate.com>

Date: Wed Dec 10 14:43:42 2014 +0530

New branch update

commit d7cf814ac2fe33ba0a5c8061cb3600d1d51e0ebb

Author: Hema <Hema.g@igate.com>

Date: Mon Dec 8 12:07:49 2014 +0530

s Please enter the commit message for your changes. Lines starting

version 02:wq

:

commit 2af2ac14bb21309e33d7f1b6bbab08059d639274

Author: Hema <Hema.g@igate.com>

Date: Mon Dec 8 10:33:43 2014 +0530

W

D:\gitDir>git merge Testing

Already up-to-date.

D:\gitDir>git diff

WARNING: terminal is not fully functional

- (press RETURN)

D:\gitDir>gitk

'gitk' is not recognized as an internal or external command,

operable program or batch file.

D:\gitDir>git branch -d Testing

error: Cannot delete the branch 'Testing' which you are currently on.

D:\gitDir>git branch

\* Testing

master

D:\gitDir>git branch master

fatal: A branch named 'master' already exists.

D:\gitDir>git checkout master

Switched to branch 'master'

D:\gitDir>git branch -d Testing

error: The branch 'Testing' is not fully merged.

If you are sure you want to delete it, run 'git branch -D Testing'.

D:\gitDir>git merge Testing

Updating d7cf814..70dc4df

Fast-forward

page2.txt | 1 +

1 file changed, 1 insertion(+)

D:\gitDir>git branch -d Testing

Deleted branch Testing (was 70dc4df).

**GIT Clone and Pull:**

D:\gitDir>git branch

\* master

D:\gitDir>git branch

\* master

D:\gitDir>git add newRepo/

D:\gitDir>git commit

[master 914fc9f] file changed

1 file changed, 4 insertions(+)

D:\gitDir>git clone D:\newGITDir repo1

Cloning into 'repo1'...

done.

D:\gitDir>git add repo1/

D:\gitDir>git commit -a

Aborting commit due to empty commit message.

D:\gitDir>git commit -a

[master 17cbd07] v1

4 files changed, 13 insertions(+)

create mode 100644 repo1/file1.txt

create mode 100644 repo1/file2.txt

create mode 100644 repo1/file3.txt

create mode 100644 repo1/file4.txt

D:\gitDir>

D:\newGITDir>git fetch D:\gitDir master

warning: no common commits

remote: Counting objects: 25, done.

remote: Compressing objects: 100% (16/16), done.

remote: Total 25 (delta 7), reused 0 (delta 0)

Unpacking objects: 100% (25/25), done.

From D:\gitDir

\* branch master -> FETCH\_HEAD

D:\newGITDir>git pull D:\gitDir master

From D:\gitDir

\* branch master -> FETCH\_HEAD

Merge made by the 'recursive' strategy.

newRepo/file1.txt | 4 ++++

newRepo/file2.txt | 1 +

newRepo/file3.txt | 5 +++++

newRepo/file4.txt | 4 ++++

page1.txt | 4 ++++

page2.txt | 2 ++

page3.txt | 1 +

page4.txt | 5 +++++

repo1/file1.txt | 4 ++++

repo1/file2.txt | 1 +

repo1/file3.txt | 1 +

repo1/file4.txt | 7 +++++++

12 files changed, 39 insertions(+)

create mode 100644 newRepo/file1.txt

create mode 100644 newRepo/file2.txt

create mode 100644 newRepo/file3.txt

create mode 100644 newRepo/file4.txt

create mode 100644 page1.txt

create mode 100644 page2.txt

create mode 100644 page3.txt

create mode 100644 page4.txt

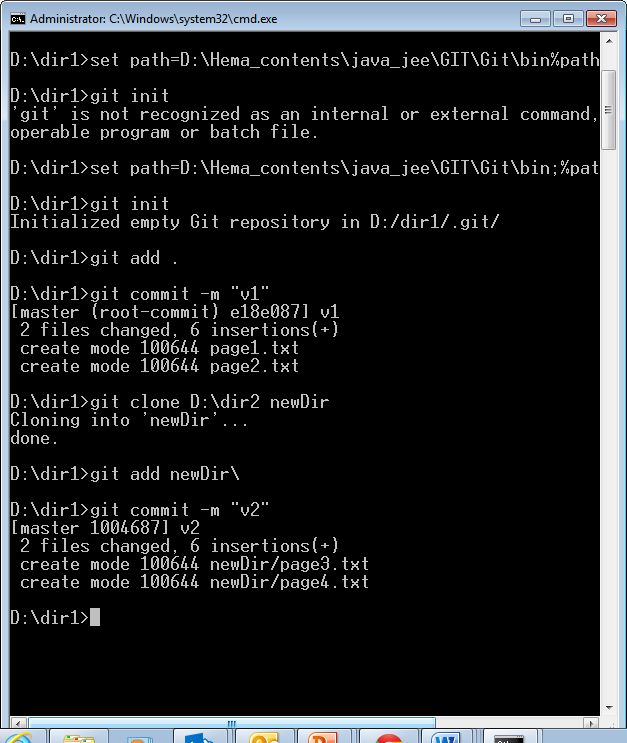
create mode 100644 repo1/file1.txt

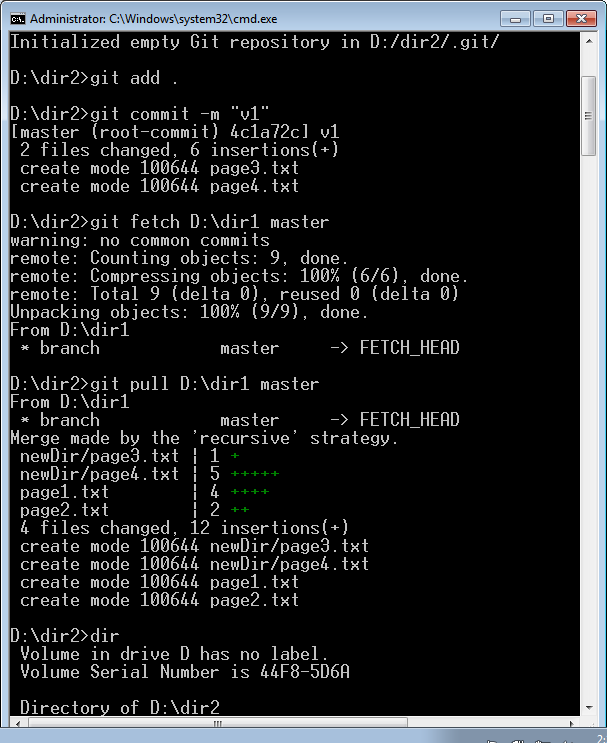
create mode 100644 repo1/file2.txt

create mode 100644 repo1/file3.txt

create mode 100644 repo1/file4.txt

D:\newGITDir>





**GIT PUSH:**

**Demo1:**

C:\Users\ghemagh>D:

D:\>mkdir test

D:\>cd test

D:\test>mkdir A

D:\test>mkdir B

D:\test>mkdir C

D:\test>mkdir D

D:\test>set path=D:\Hema\_contents\java\_jee\GIT\Git\bin;%path%

D:\test>cd A

D:\test\A>git init --bare

Initialized empty Git repository in D:/test/A/

D:\test\A>cd..

D:\test>cd B

D:\test\B>git init

Initialized empty Git repository in D:/test/B/.git/

D:\test\B>cd..

D:\test>cd C

D:\test\C>git init

Initialized empty Git repository in D:/test/C/.git/

D:\test\C>cd..

D:\test>cd D

D:\test\D>git init

Initialized empty Git repository in D:/test/D/.git/

D:\test\D>cd ..

D:\test>git clone --bare /D:/test/A /D:/test/E

fatal: could not create leading directories of '/D:/test/E'

D:\test>git clone --bare /d/test/A/d/test/E

fatal: repository '/d/test/A/d/test/E' does not exist

D:\test>git clone --bare d:/test/A d:/test/E

Cloning into bare repository 'd:/test/E'...

warning: You appear to have cloned an empty repository.

done.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Demo 2:**

D:\test>cd B

D:\test\B>touch info.txt

D:\test\B>echo "Project">info.txt

D:\test\B>git add info.txt

D:\test\B>git status

On branch master

Initial commit

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: info.txt

D:\test\B>git commit info.txt -m "for admin"

[master (root-commit) 96e3fe2] for admin

1 file changed, 1 insertion(+)

create mode 100644 info.txt

D:\test\B>git branch Development

D:\test\B>git branch Testing

D:\test\B>git branch

Development

Testing

\* master

D:\test\B>git push C:\test\A Development

fatal: 'C:\test\A' does not appear to be a git repository

fatal: Could not read from remote repository.

Please make sure you have the correct access rights

and the repository exists.

D:\test\B>git push D:\test\A Development

Counting objects: 3, done.

Writing objects: 100% (3/3), 214 bytes | 0 bytes/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To D:\test\A

\* [new branch] Development -> Development

D:\test\B>git push D:\test\A Testing

Total 0 (delta 0), reused 0 (delta 0)

To D:\test\A

\* [new branch] Testing -> Testing

**Creating Tags:**

D:\dir1>git tag version1.6 -m "version 1.6"

D:\dir1>git tag

version1.6

D:\dir1>git commit -m "v3"

[master cbbd0e4] v3

6 files changed, 12 insertions(+), 36 deletions(-)

delete mode 100644 RecClient.java

delete mode 100644 ReceiverServer.java

create mode 100644 newDir/page3.txt

create mode 100644 newDir/page4.txt

create mode 100644 page1.txt

create mode 100644 page2.txt

D:\dir1>git tag version 1.7 commit v3

fatal: too many params

D:\dir1>git tag version 1.7 commit "v3"

fatal: too many params

D:\dir1>git tag version1.7 commit "v3"

fatal: too many params

D:\dir1>git tag version1.7 -m "version1.7" commit "v3"

fatal: too many params

D:\dir1>git tag version1.7 -m "version1.7" "v3"

fatal: Failed to resolve 'v3' as a valid ref.

D:\dir1>git tag version1.7 -m "version1.7" cbbd0e4

D:\dir1>cat page2.txt

Happy Weekend

D:\dir1>git commit -m "v4"

On branch master

Changes not staged for commit:

modified: page2.txt

no changes added to commit

D:\dir1>git page2.txt commit -m "v4"

git: 'page2.txt' is not a git command. See 'git --help'.

D:\dir1>git commit page2.txt -m "v4"

[master d09b46a] v4

1 file changed, 1 insertion(+), 2 deletions(-)

D:\dir1>cat page2.txt

Happy Weekend

D:\dir1>git checkout version1.7

Note: checking out 'version1.7'.

You are in 'detached HEAD' state. You can look around, make experimental

changes and commit them, and you can discard any commits you make in this

state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may

do so (now or later) by using -b with the checkout command again. Example:

git checkout -b new\_branch\_name

HEAD is now at cbbd0e4... v3

D:\dir1>cat page2.txt

Welcome to IGATE's workplace

\* The new 6 digit id is to be used

D:\dir1>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**GIT Rebase:**

D:\dir1>git checkout -b rebasetest

Switched to a new branch 'rebasetest'

D:\dir1>touch rebase1.txt

D:\dir1>git add . && git commit -m "v1"

[rebasetest 2767009] v1

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 rebase1.txt

D:\dir1>git checkout master

Switched to branch 'master'

D:\dir1>echo Writing to file > rebasefile.txt

D:\dir1>git add rebasefile.txt

D:\dir1>git commit -m "New file"

[master 47c06d3] New file

1 file changed, 1 insertion(+)

create mode 100644 rebasefile.txt

D:\dir1>git checkout rebasetest

Switched to branch 'rebasetest'

D:\dir1>git rebase master

First, rewinding head to replay your work on top of it...

Applying: v1

D:\dir1>git checkout master

Switched to branch 'master'

D:\dir1>git merge rebasetest

Updating 47c06d3..7a6e739

Fast-forward

rebase1.txt | 0

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 rebase1.txt

D:\dir1>dir

Volume in drive D has no label.

Volume Serial Number is 44F8-5D6A

Directory of D:\dir1

12/12/2014 11:47 AM <DIR> .

12/12/2014 11:47 AM <DIR> ..

12/11/2014 02:20 PM <DIR> newDir

12/12/2014 10:55 AM 43 page1.txt

12/12/2014 10:55 AM 15 page2.txt

12/12/2014 11:47 AM 0 rebase1.txt

12/12/2014 11:47 AM 18 rebasefile.txt

4 File(s) 76 bytes

3 Dir(s) 69,680,730,112 bytes free

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Remotes:

Adding a remote

To add a new remote, use the git remote add command on the terminal, in the directory your repository is stored at.

The git remote add command takes two arguments:

* A remote name, for example, origin
* A remote URL, for example, https://github.com/user/repo.git

For example:

git remote add origin https://github.com/*user*/*repo*.git

# Set a new remote

git remote -v

# Verify new remote

# origin https://github.com/*user*/*repo*.git (fetch)

# origin https://github.com/*user*/*repo*.git (push)

## Pushing to a remote

Use git push to push your local branch to a remote repository.

The git push command takes two arguments:

* A remote name, for example, origin
* A branch name, for example, master

For example:

git push <REMOTENAME> <BRANCHNAME>

As an example, you usually run git push origin master to push your local changes to your online repository.

### Renaming branches

To rename a branch, you'd use the same git push command, but you would add one more argument: the name of the new branch. For example:

git push <REMOTENAME> <LOCALBRANCHNAME>:<REMOTEBRANCHNAME>

This pushes the LOCALBRANCHNAME to your REMOTENAME, but it is renamed to REMOTEBRANCHNAME.

### Dealing with "non-fast-forward" errors

If your local copy of a repository is out of sync with, or "behind," the upstream repository you're pushing to, you'll get a message saying non-fast-forward updates were rejected. This means that you must retrieve, or "fetch," the upstream changes, before you are able to push your local changes.

For more information on this error, see "[Dealing with non-fast-forward errors](https://help.github.com/articles/dealing-with-non-fast-forward-errors)."

### Pushing tags

By default, and without additional parameters, git push sends all matching branches that have the same names as remote branches.

To push a single tag, you can issue the same command as pushing a branch:

git push <REMOTENAME> <TAGNAME>

To push all your tags, you can type the command:

git push <REMOTENAME> --tags

### Deleting a remote branch or tag

The syntax to delete a branch is a bit arcane at first glance:

git push <REMOTENAME> :<BRANCHNAME>

Note that there is a space before the colon. The command resembles the same steps you'd take to rename a branch. However, here, you're telling Git to push nothing into BRANCHNAME on REMOTENAME. Because of this, git push deletes the branch on the remote repository.

### Remotes and forks

You might already know that [you can "fork" repositories](https://guides.github.com/overviews/forking/) on GitHub.

When you clone a repository you own, you provide it with a remote URL that tells Git where to fetch and push updates. If you want to collaborate with the original repository, you'd add a new remote URL, typically called upstream, to your local Git clone:

git remote set-url upstream <THEIR\_REMOTE\_URL>

Now, you can fetch updates and branches from their fork:

git fetch upstream branch

# Grab the upstream remote's branches

# remote: Counting objects: 75, done.

# remote: Compressing objects: 100% (53/53), done.

# remote: Total 62 (delta 27), reused 44 (delta 9)

# Unpacking objects: 100% (62/62), done.

# From https://github.com/octocat/repo

# \* [new branch] master -> upstream/master

When you're done making local changes, you can push your local branch to GitHub and [initiate a pull request](https://help.github.com/articles/using-pull-requests).