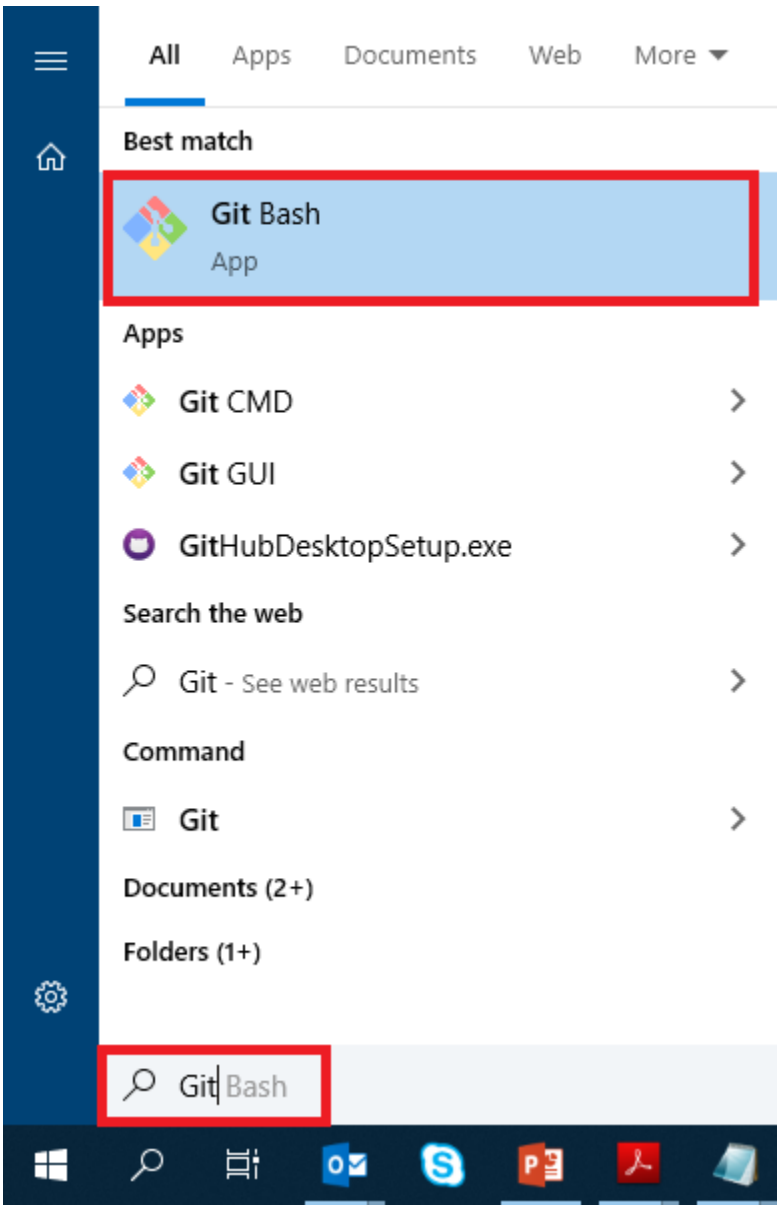
Table of content	**Page Number**
Basic Git commands and Configuring Git	2
Creating Local Repository in Git	4
Adding file into local repository	6
Creating repository into the existing directory	9
Modifying the files within repository	12
Creating your first remote repository on Github	14
Mapping your remote repository on Github with local repository	16
Modifying the files in local repository and reflecting the changes into the GitHub repositopry	19
Updating Github repository and receiving changes into local repository	21
Cloning your own Github repository	24
Cloning other users Github repository	27
Branching the local repository	37
Viewing the write history	40

-: Basic Git commands and Configuring Git :-

Open GitBash in Windows:



Checking installed version of Git:

```
$ git --version
git version 2.27.0.windows.1
```

Configuring Username and password:

```
$ git config --global user.name "Manish Rajhans"
```

```
$ git config --global user.email "manish.rajhans@capgemini.com"
```

After configuring username and password, verify it with below commands:

```
$ git config --global user.name
Manish Rajhans
```

```
$ git config --global user.email
manish.rajhans@capgemini.com
```

Obtaining help from Git:

```
$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone             Clone a repository into a new directory
   init              Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add               Add file contents to the index
   mv                Move or rename a file, a directory, or a symlink
   restore           Restore working tree files
   rm                Remove files from the working tree and from the index
   sparse-checkout   Initialize and modify the sparse-checkout
```

- You can also use `git help <commandName>` to receive the browser based help.

-: Creating Local Repository in Git :-

Step1. Check the current folder.

```
$ pwd
/c/Users/mrajhans
```

Step2. Now create some folder in which you want to add repository and navigate to it. In our case, we have already created GitRepos folder and now navigate to it.

```
$ cd c:/study/GitRepos
```

Step3. In GitRepos folder lets create one directory and navigate to it.

```
$ mkdir HelloWorld
```

```
$ cd HelloWorld/
```

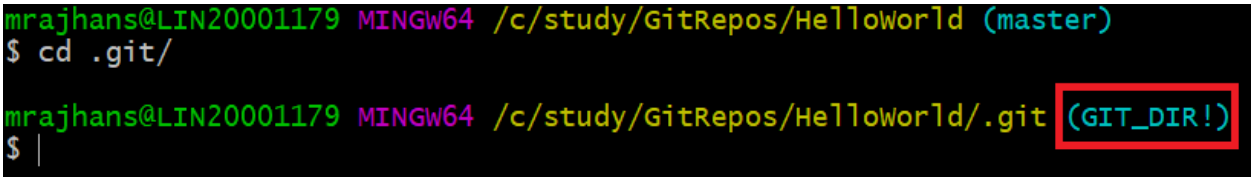
Step4. To initialize the git repository in this directory us git init command as below.

```
$ git init
Initialized empty Git repository in C:/Study/GitRepos/HelloWorld/.git/
```

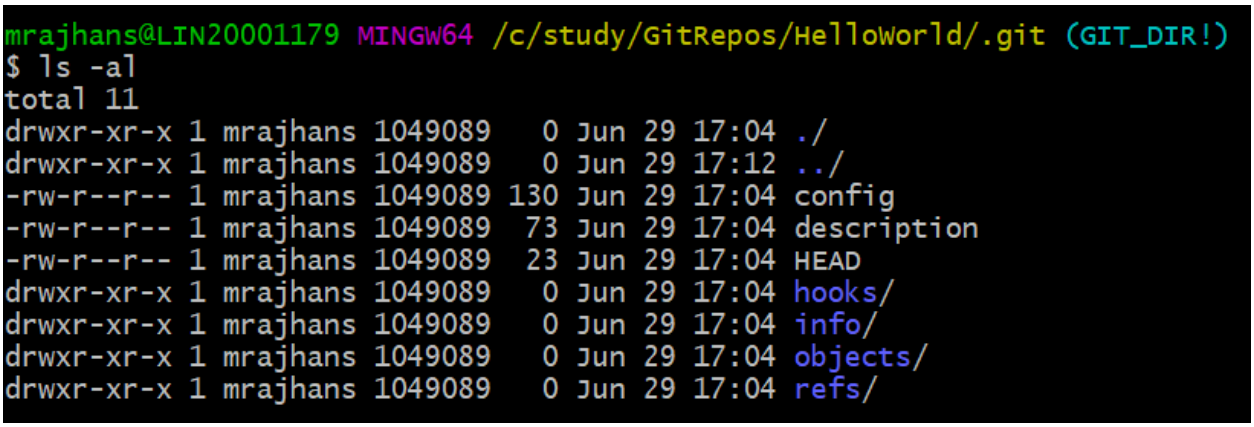
Step5. Now observe the prompt, you can see it shows it as master branch.

```
mrajhans@LIN20001179 MINGW64 /c/study/GitRepos/HelloWorld (master)
$
```

Step6. Whenever you have run the git init, it has installed the .git directory in the repository folder. It indicates the git databse used for maintaining the commit history etc. You can view it as below:



Step7. You can also navigate and check the .git directory as below.



-: Adding file into local repository:-

Step1. Navigate to the HelloWorld directory (if already not there).

```
mrajhans@LIN20001179 MINGW64 /c/study/GitRepos/HelloWorld (master)
```

Step2. Create a file in the current directory.

```
$ nano helloworld.js
```

Step3. This will open the file in edit mode, as shown below. You can write anything into it.

Press Ctrl+X to close and press Y to save changes and filename will appear, then hit enter.

Step4. After adding the file, you can see that git can track the status of file.

```
$ git status
On branch master

No commits yet

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

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

Step5. This file is currently in the wotking directory. To add it into the staging area, we can use git add command.

```
mrajhans@LIN20001179 MINGW64 /c/study/GitRepos/HelloWorld (master)
$ git add helloworld.js
warning: LF will be replaced by CRLF in helloworld.js.
The file will have its original line endings in your working directory
```

Step6. Now once again we can check the status and it will show that new file got added. Earlier file created was untracked as it wasn't added into the staging area.

```
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   helloworld.js
```

Step7. You can also remove the file from the staging area and put it back into the working directory, you can use git reset command for it.

```
$ git status
On branch master

No commits yet

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

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

Step8. For demonstration purpose, we will add the file again, to add the file again in staging area, use git add command.

Step9. Now this added file into staging area can be written to repository with commit command.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/HelloWorld (master)
$ git commit -m "First Commit"
[master (root-commit) 83edd88] First Commit
 1 file changed, 1 insertion(+)
 create mode 100644 helloworld.js
```

Step10. Once again we will check the status of the repository as below.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/HelloWorld (master)
$ git status
On branch master
nothing to commit, working tree clean
```

-: Creating repository into the existing directory:-

Step1. You can download the sample repository from material shared with you. Unzip it and copy it to the location you wanted to and navigate to it using GitBash.

```
mrajhans@LIN20001179 MINGW64 ~
$ cd C:/Study/GitRepos/SampleRepository
```

Step2. Now we will initialize the directoy with repository.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository
$ git init
Initialized empty Git repository in C:/Study/GitRepos/SampleRepository/.git/
```

You can see it has added .git directory which means that it has created the database needed to maintain the changes into the repository.

Step3. Lets check the status of the newly created repository. It will show the list of files and folder untracked yet.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        404.html
        apple-touch-icon.png
        browserconfig.xml
        css/
        favicon.ico
        humans.txt
        index.html
        js/
        tile-wide.png
        tile.png

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

Step4. Now add all the existing files from the working directory into staging area with git add command.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git add .
warning: LF will be replaced by CRLF in 404.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in browserconfig.xml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/main.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in css/normalize.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in humans.txt.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in js/main.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in js/vendor/jquery-1.11.2.min.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in js/vendor/modernizr-2.8.3-respond-1.4.2.min.js.
The file will have its original line endings in your working directory
```

Step5. After getting all files added, check the status of repository, it will show the status as below.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   404.html
        new file:   apple-touch-icon.png
        new file:   browserconfig.xml
        new file:   css/main.css
        new file:   css/normalize.css
        new file:   css/normalize.min.css
        new file:   favicon.ico
        new file:   humans.txt
        new file:   index.html
        new file:   js/main.js
        new file:   js/vendor/jquery-1.11.2.min.js
        new file:   js/vendor/modernizr-2.8.3-respond-1.4.2.min.js
        new file:   tile-wide.png
        new file:   tile.png
```

Step6. After files added into the staging area, commit the changes to the local repository.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git commit -m "First Commit"
[master (root-commit) 1bd1941] First Commit
 14 files changed, 992 insertions(+)
 create mode 100644 404.html
 create mode 100644 apple-touch-icon.png
 create mode 100644 browserconfig.xml
 create mode 100644 css/main.css
 create mode 100644 css/normalize.css
 create mode 100644 css/normalize.min.css
 create mode 100644 favicon.ico
 create mode 100644 humans.txt
 create mode 100644 index.html
 create mode 100644 js/main.js
 create mode 100644 js/vendor/jquery-1.11.2.min.js
 create mode 100644 js/vendor/modernizr-2.8.3-respond-1.4.2.min.js
 create mode 100644 tile-wide.png
 create mode 100644 tile.png
```

Step7. Commit will commit the changes to the repository, hence now check the status again.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git status
On branch master
nothing to commit, working tree clean
```

-: Modifying the files within repository:-

Step1. Open the file and modify some content from the index.html file. (I assume you know how to store and come out on the command prompt)

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ nano index.html
```

Step2. Now if you check the status, you will see that the status of file it shows as modified.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git status
On branch master
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:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
```

Step3. Now move the modified into from working directory to staging area.

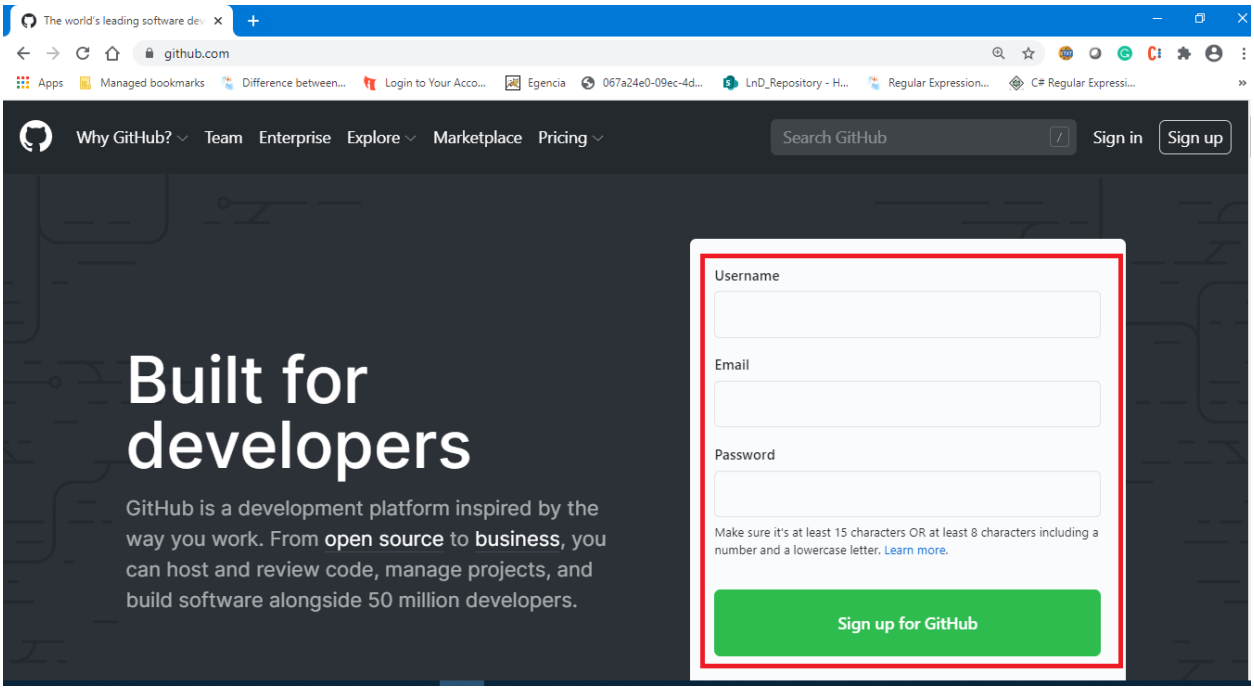
```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git add .
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory
```

Step4. Now commit the changes into the repository.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git commit -m "Updated index file commit"
[master d514e91] Updated index file commit
 1 file changed, 1 insertion(+), 1 deletion(-)
```

-: Creating remote repository on Github :-

Step1. Navigate to Github.com and register on the site.

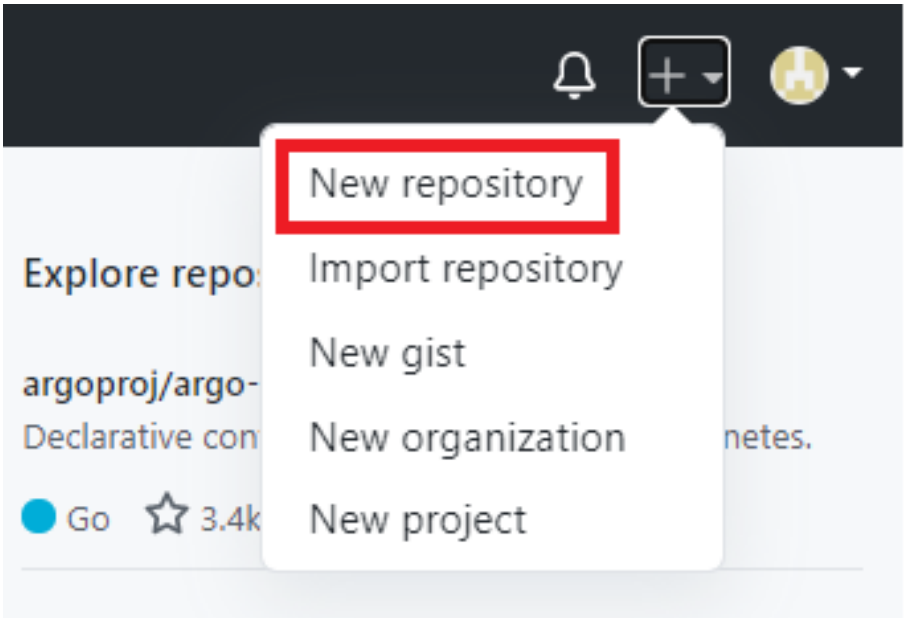


Step2. After successful registration, you will receive an email to verify your email address. Click on Verify Button.

Step3. That's it. Your account is ready to be used.

-: Creating your first remote repository on Github :-

Step1. After email verification is successful as shown in the previous step, now sign in to the Github account and after signing in, you should able to see something like below in topright corner.



Click on + sign and then click on New Repository.

Step2. After clicking on the New repository, below page will open. In that page specify repository name, it will check for availability and if available, as below it will show green colored tick. We have option to create public and private repositories. Now for this example, we will create public repository. And then you can click on Create repository.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
Import a repository.

Owner *

manishrajhans

Repository name *

samplerepository

samplerepository is available.

Great repository names are s spiration? How about super-goggles?

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer.

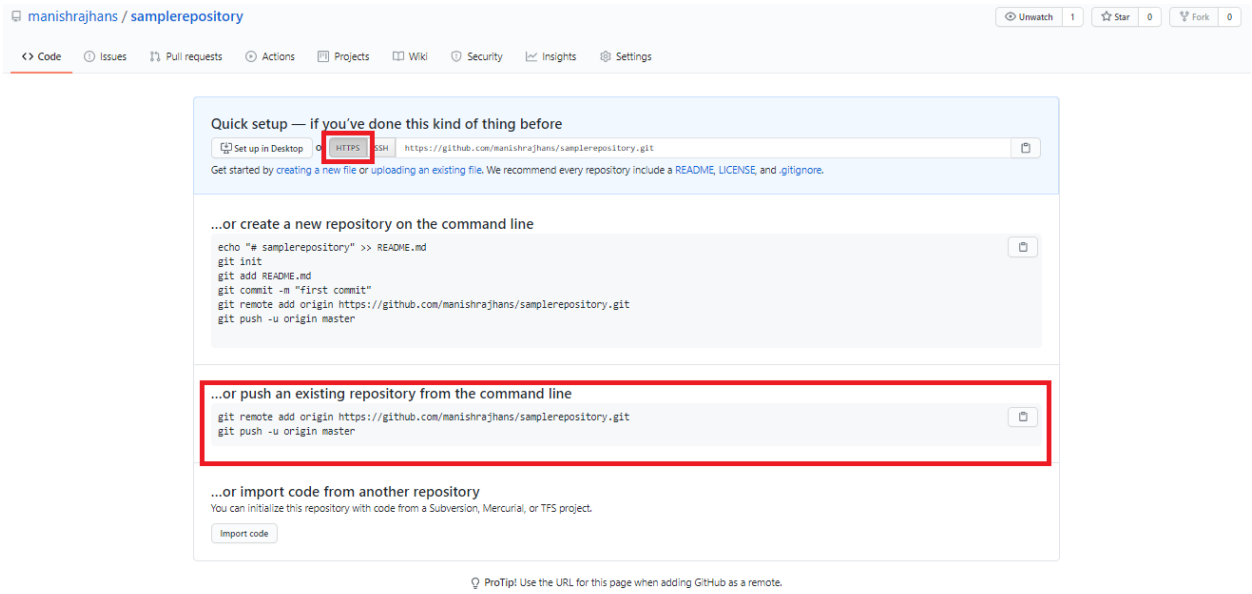
Add .gitignore: None

Add a license: None

Create repository

-: Mapping your remote repository on Github with local repository:-

Step1. After clicking on Create repository, you will get option to map your local repository with the remote repository. Here ensure that you will select HTTPS option. As shown below the highlighted commands we can use to map our local repository with the remote repository.



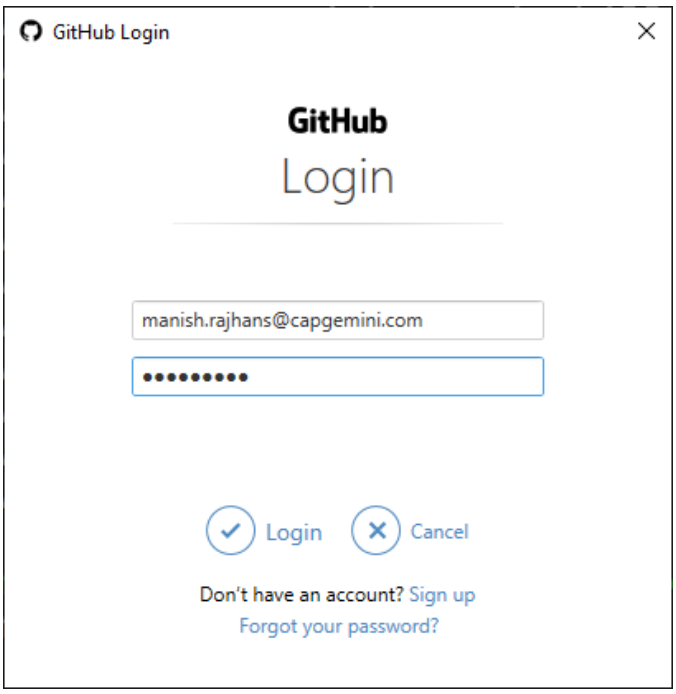
Step2. Comeback to GitBash and copy the first command from the highlighted section into local repository.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git remote add origin https://github.com/manishrajhans/samplerepository.git
```

Step3. After mapping your local repository with the remote repository, we would push the changes committed to the local repository to the remote repository.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git push -u origin master
```

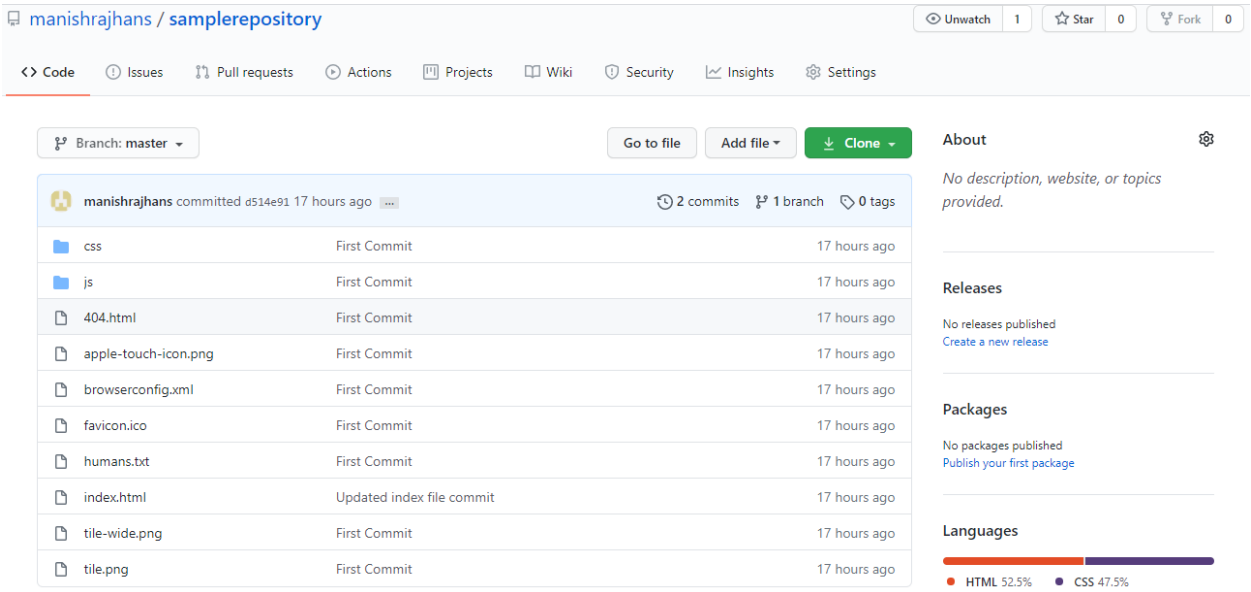
After you run the push command, for the first time , you will be prompted for Github credentials.



After specifying the credentials all the committed changes from local repository will be written to the remote repository. (Note for pushing changes, credentials will be needed only for the first time).

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git push -u origin master
error: waitpid for git credential-manager get failed: No child processes
error: waitpid for git credential-manager store failed: No child processes
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 4 threads
Compressing objects: 100% (21/21), done.
Writing objects: 100% (22/22), 57.98 KiB | 1.29 MiB/s, done.
Total 22 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
error: waitpid for send-pack failed: No child processes
To https://github.com/manishrajhans/samplerepository.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

Step4. After this if you refresh the Github repository, you will see all the changes from local repository are now also available in Github repository.


manishrajhans / samplerepository
Unwatch 1
Star 0
Fork 0
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Branch: master
Go to file
Add file
Clone
manishrajhans committed d514e91 17 hours ago ...
2 commits
1 branch
0 tags
css First Commit 17 hours ago
js First Commit 17 hours ago
404.html First Commit 17 hours ago
apple-touch-icon.png First Commit 17 hours ago
browserconfig.xml First Commit 17 hours ago
favicon.ico First Commit 17 hours ago
humans.txt First Commit 17 hours ago
index.html Updated index file commit 17 hours ago
tile-wide.png First Commit 17 hours ago
tile.png First Commit 17 hours ago
About
No description, website, or topics provided.
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
HTML 52.5%
CSS 47.5%

-: Modifying the files in local repository and reflecting the changes into the GitHub repositopry:-

Step1. Modify the index.html file. And also check the status. It will show the file is modified however not yet added into staging area as below.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ nano index.html

mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

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:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
```

Step2. Add the changes from working directory to the staging area.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git add .
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory
```

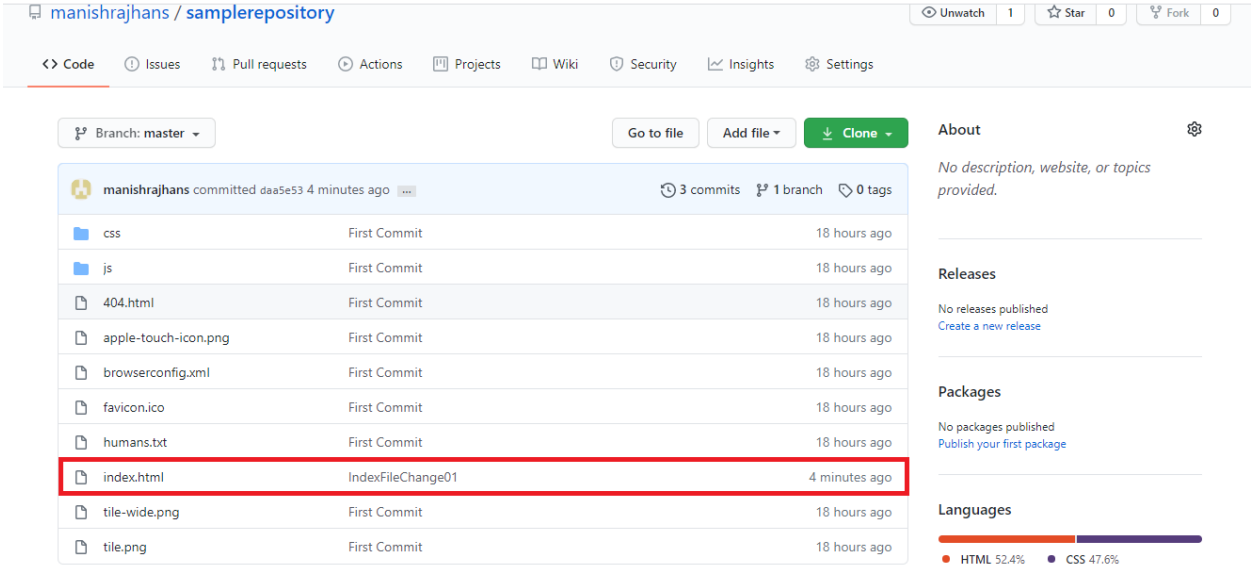
Step3. Now commit the changes to the local repository.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git commit -m "IndexFileChange01"
[master daa5e53] IndexFileChange01
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Step4. Push the changes from local repository to the Github repository. This time it will not ask for any credentials.

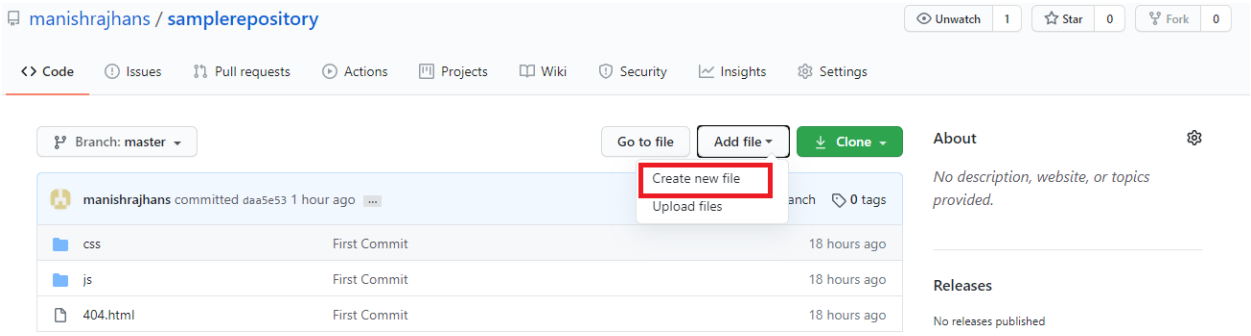
```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git push
error: waitpid for git credential-manager get failed: No child processes
error: waitpid for git credential-manager store failed: No child processes
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 304 bytes | 152.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
error: waitpid for send-pack failed: No child processes
To https://github.com/manishrajhans/samplerepository.git
   d514e91..daa5e53  master -> master
```

Step 5. After the push command is executed successfully. View the changes into the Github repository.

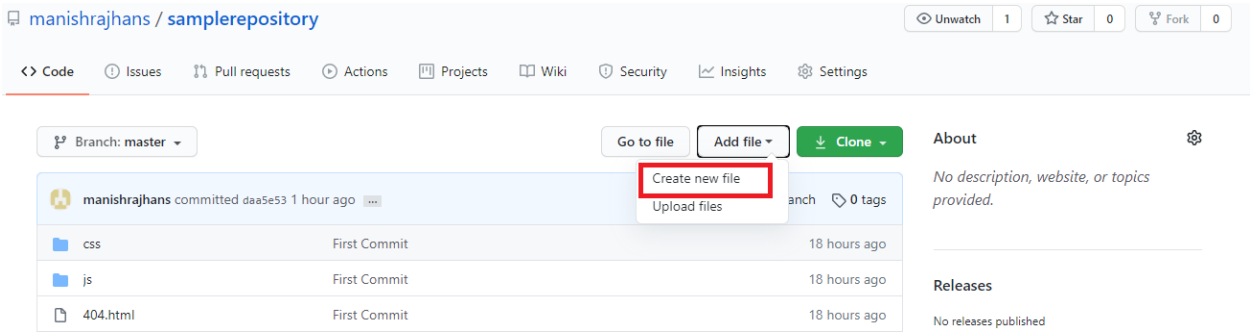


-: Updating Github repository and receiving changes into local repository:-

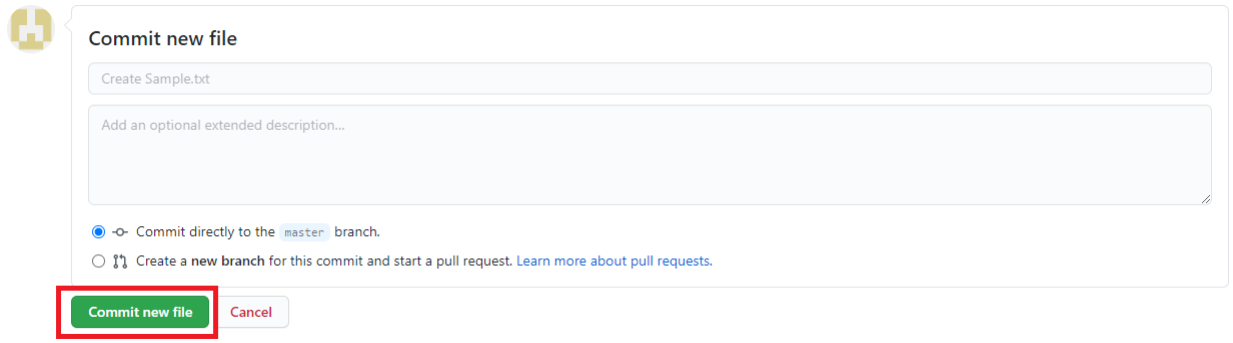
Step1. You can also add the files directly from the GitHub site UI either by creating it or uploading it, to add the files from Github click on Add File, then click on Create new file



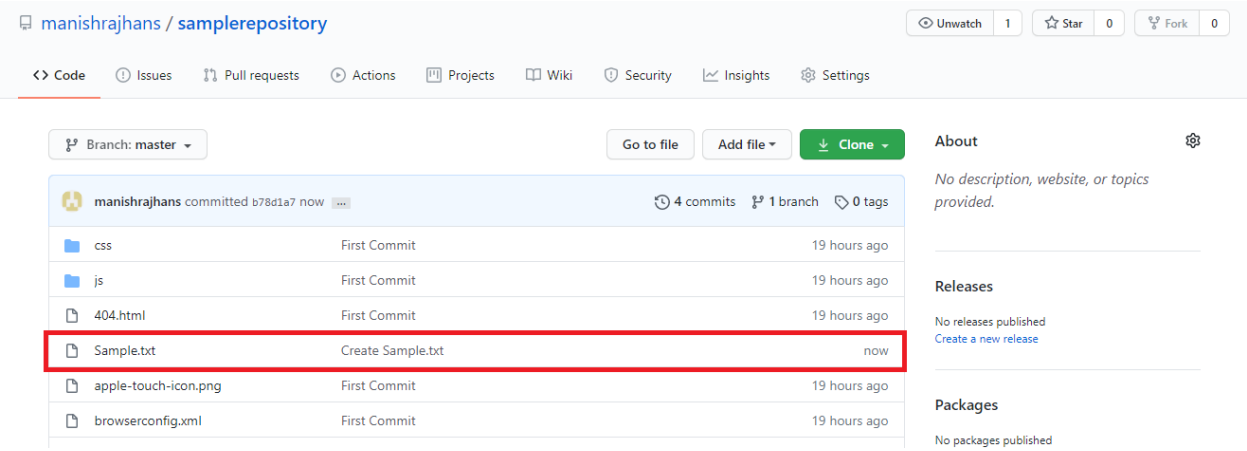
Step2. Name the file and add some content into it.



Step3. Now to add this file into the repository, click on Commit New File button down below.



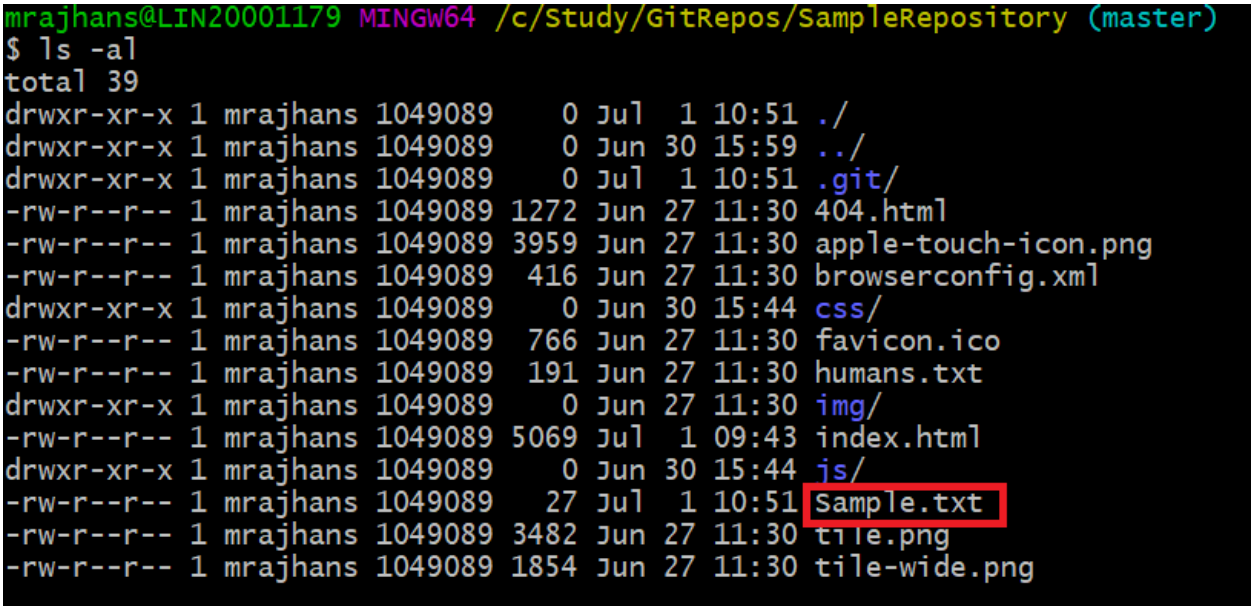
Step4. After commiting the file, you can check the repository and you will see the changes.



Step5. Pull the changes now from github repository to local repository.

```
mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ git pull
From https://github.com/manishrajhans/samplerepository
   daa5e53..b78d1a7  master     -> origin/master
Updating daa5e53..b78d1a7
Fast-forward
 Sample.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 Sample.txt
```

Step6. Now check the status of the local repository to check whether the local repository is added with the file updated from the GitHub repository or not.


mrajhans@LIN20001179 MINGW64 /c/Study/GitRepos/SampleRepository (master)
$ ls -al
total 39
drwxr-xr-x 1 mrajhans 1049089   0 Jul 1 10:51 ./
drwxr-xr-x 1 mrajhans 1049089   0 Jun 30 15:59 ../
drwxr-xr-x 1 mrajhans 1049089   0 Jul 1 10:51 .git/
-rw-r--r-- 1 mrajhans 1049089 1272 Jun 27 11:30 404.html
-rw-r--r-- 1 mrajhans 1049089 3959 Jun 27 11:30 apple-touch-icon.png
-rw-r--r-- 1 mrajhans 1049089 416 Jun 27 11:30 browserconfig.xml
drwxr-xr-x 1 mrajhans 1049089   0 Jun 30 15:44 css/
-rw-r--r-- 1 mrajhans 1049089 766 Jun 27 11:30 favicon.ico
-rw-r--r-- 1 mrajhans 1049089 191 Jun 27 11:30 humans.txt
drwxr-xr-x 1 mrajhans 1049089   0 Jun 27 11:30 img/
-rw-r--r-- 1 mrajhans 1049089 5069 Jul 1 09:43 index.html
drwxr-xr-x 1 mrajhans 1049089   0 Jun 30 15:44 js/
-rw-r--r-- 1 mrajhans 1049089   27 Jul 1 10:51 Sample.txt
-rw-r--r-- 1 mrajhans 1049089 3482 Jun 27 11:30 tile.png
-rw-r--r-- 1 mrajhans 1049089 1854 Jun 27 11:30 tile-wide.png

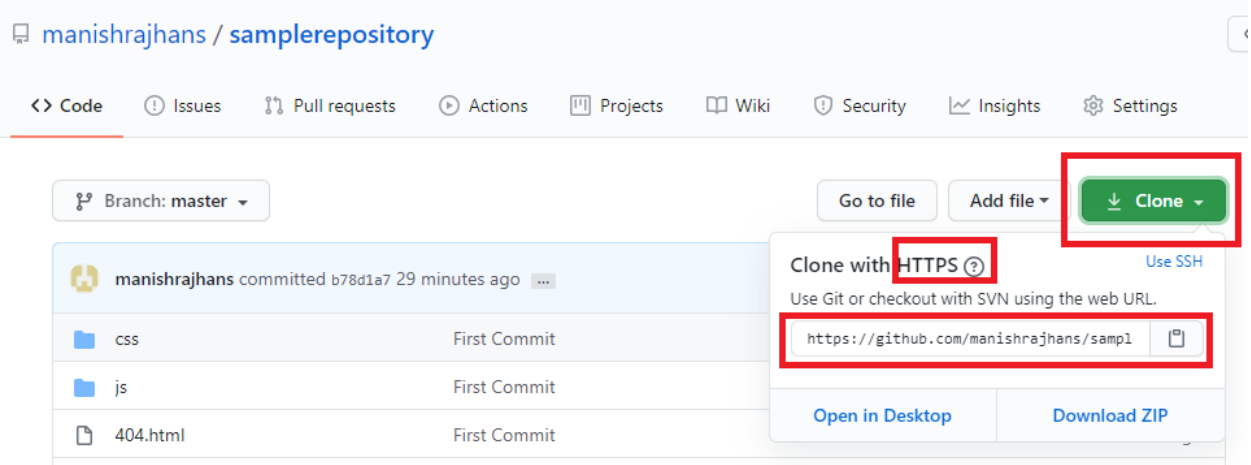
-: Cloning your own Github repository:-

Step1. Create new directory in which you want to clone the repository and navigate to it.

```
mrajhans@LIN20001179 MINGW64 /c/study
$ mkdir ClonedRepo

mrajhans@LIN20001179 MINGW64 /c/study
$ cd ClonedRepo
```

Step2. Now navigate to your Github repository. Click on Clone button. Ensure clone with HTTPS is selected. Now copy the URL from it, as shown below.



Step3. Come back to Gitbash and now type the command git clone with copied URL as shown below. (Incase while running this command if you get the error.)

```
$ git clone https://github.com/manishrajhans/samplerepository.git
Cloning into 'samplerepository'...
remote: Enumerating objects: 26, done.
remote: Counting objects: 100% (26/26), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 26 (delta 4), reused 21 (delta 2), pack-reused 0
Unpacking objects: 100% (26/26), 58.70 KiB | 279.00 KiB/s, done.
```

After this commands successful execution repo will be cloned into your current folder.

Step4. Check the cloned repository as below.

```
mrajhans@LIN20001179 MINGW64 /c/Study/ClonedRepo
$ cd samplerepository/

mrajhans@LIN20001179 MINGW64 /c/Study/ClonedRepo/samplerepository (master)
$ ls -al
total 39
drwxr-xr-x 1 mrajhans 1049089    0 Jul  2 22:11 ./
drwxr-xr-x 1 mrajhans 1049089    0 Jul  2 22:11 ../
drwxr-xr-x 1 mrajhans 1049089    0 Jul  2 22:11 .git/
-rw-r--r-- 1 mrajhans 1049089 1332 Jul  2 22:11 404.html
-rw-r--r-- 1 mrajhans 1049089 3959 Jul  2 22:11 apple-touch-icon.png
-rw-r--r-- 1 mrajhans 1049089  428 Jul  2 22:11 browserconfig.xml
drwxr-xr-x 1 mrajhans 1049089    0 Jul  2 22:11 css/
-rw-r--r-- 1 mrajhans 1049089  766 Jul  2 22:11 favicon.ico
-rw-r--r-- 1 mrajhans 1049089  206 Jul  2 22:11 humans.txt
-rw-r--r-- 1 mrajhans 1049089 5157 Jul  2 22:11 index.html
drwxr-xr-x 1 mrajhans 1049089    0 Jul  2 22:11 js/
-rw-r--r-- 1 mrajhans 1049089   27 Jul  2 22:11 Sample.txt
-rw-r--r-- 1 mrajhans 1049089 3482 Jul  2 22:11 tile.png
-rw-r--r-- 1 mrajhans 1049089 1854 Jul  2 22:11 tile-wide.png
```

Step5. We will also now try to add files into the local repository and push the changes to the Github repository. To do this add file named sample.txt into the repository, add it into staging area, commit the changes and push the changes into the GitHub repository.

```
mrajhans@LIN20001179 MINGW64 /c/Study/ClonedRepo/samplerepository (master)
$ nano sample.txt

mrajhans@LIN20001179 MINGW64 /c/Study/ClonedRepo/samplerepository (master)
$ git add .

mrajhans@LIN20001179 MINGW64 /c/Study/ClonedRepo/samplerepository (master)
$ git commit -m "SampleFileUpdated"
[master c3cb1a6] SampleFileUpdated
 1 file changed, 1 insertion(+)

mrajhans@LIN20001179 MINGW64 /c/Study/ClonedRepo/samplerepository (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 319 bytes | 45.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/manishrajhans/samplerepository.git
   b78d1a7..c3cb1a6  master -> master
```

Step6. The added file sample.txt you can see added into the Github repository.

-: Cloning other users Github repository:-

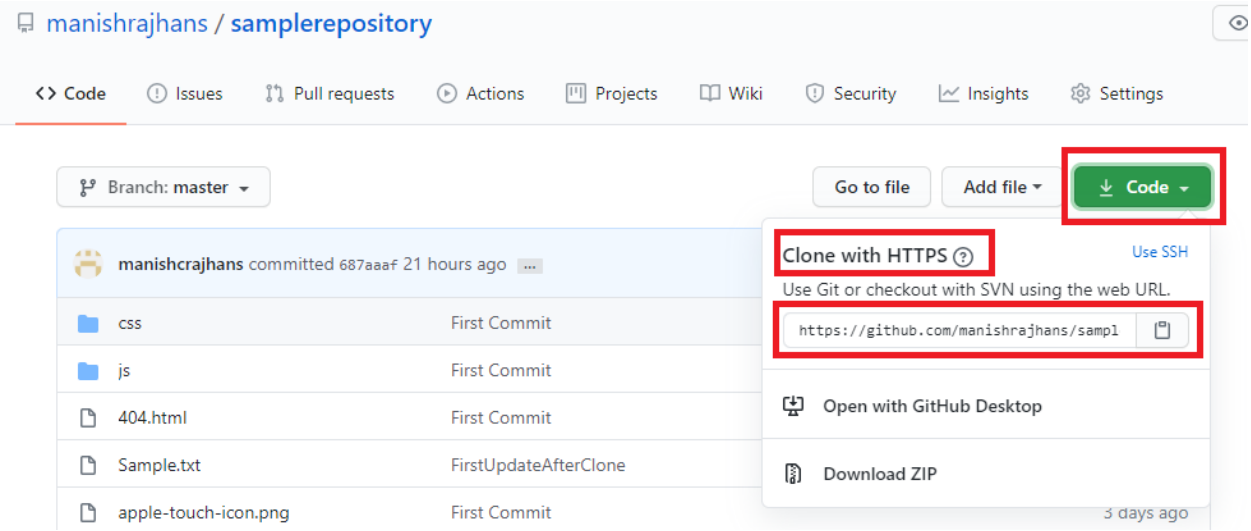
Pressumptions:

- Other User's Name: Manish Rajhans
- Other User's Email: manish.rajhans@capgemini.com
- Repository: manishrajhans/samplerepository
  (In this case I recommend to take this repository as your own repository.)

- Name: Manish R
- Email: manish.rajhans@outlook.com
- Repository: manishcrajhans/samplerepository
  (In this case I recommend to take this repository as your colleague's repository.)

  [Note: Try this collaboratively with your collegue.]

Step1. To clone the other user's repository, copy the HTTPS Url for cloning the repository.



Step2. On your machine create new directory in which you want to clone the directory and run the clone like in previous example.

```
$ git clone https://github.com/manishrajhans/samplerepository.git
Cloning into 'samplerepository'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 32 (delta 7), reused 26 (delta 3), pack-reused 0
Unpacking objects: 100% (32/32), 59.20 KiB | 31.00 KiB/s, done.
```

Step3. Now you can check that cloned sample directory is added on your machine, after checking that navigate to the directory.

```
$ cd samplerepository/
```

Step4. Now modify the changes in the sample.txt of the cloned repository.

```
$ nano sample.txt
```

Step5. Add the modified files into staging area.

```
$ git add .
```

Step6. Commit the changes to the local directory.

```
$ git commit -m "Made changes into sample file from cloned repo"
[master 395b962] Made changes into sample file from cloned repo
 1 file changed, 1 insertion(+), 1 deletion(-)
```

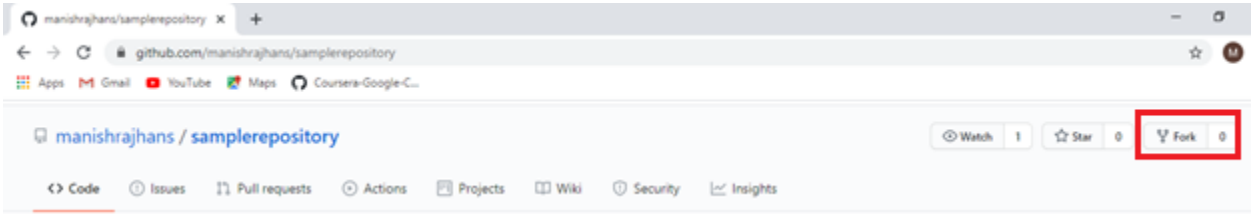
Step7. Finally push the changes to remote repository from which this repository is cloned.

```
$ git push
remote: Permission to manishrajhans/samplerepository.git denied to manishcrajhans.
fatal: unable to access 'https://github.com/manishrajhans/samplerepository.git/': The requested
URL returned error: 403
```

This step, you will get error, and the reason for this is that you are not authorized user to make the changes to the repository.

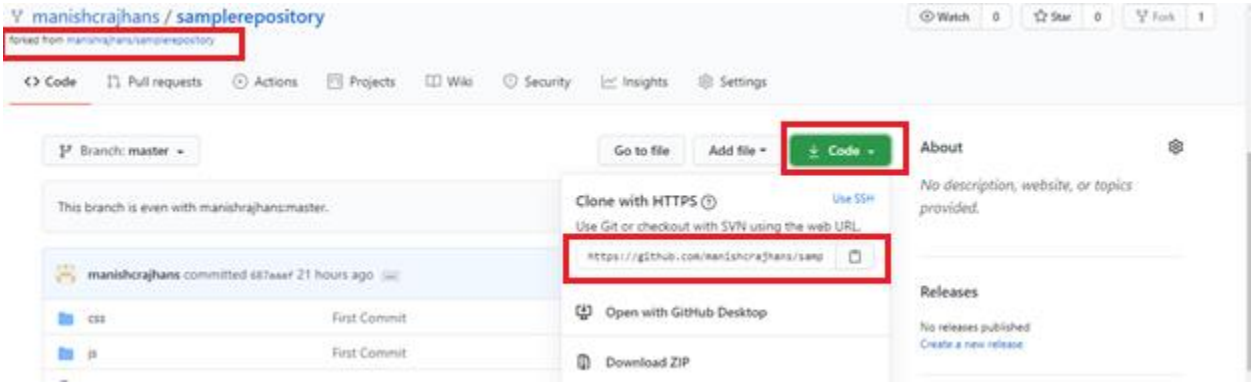
To collaborate to this repository, you need to fork the remote repository.

Step8. To fork the repository, navigate to the other users url of repository. After navigating to url, Click on fork button as shown below.



After you click on fork, it will open new page.

Step9. The newly opened page will look like this, now observe the repository name section where it shows that repository is forked from which repository, in the clone URL, you will see your username instead of owners username. (Observer carefully.)



Step10. Now create a new directory and clone the repository into it with below commands.

```
$ cd  ..
```

```
$ mkdir NewClonedRepos
```

```
$ cd NewClonedRepos/
```

```
$ git clone https://github.com/manishcrajhans/samplerepository.git
Cloning into 'samplerepository'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 32 (delta 7), reused 26 (delta 3), pack-reused 0
Unpacking objects: 100% (32/32), 59.20 KiB | 32.00 KiB/s, done.
```

This step depicts how forked repository can be cloned.

Step11. After cloning of repository, navigate to the repository.

```
$ cd samplerepository/
```

Step12. Modify the file sample.txt, add it into the staging area and then commit the changes into the local repository.

```
$ nano sample.txt
```

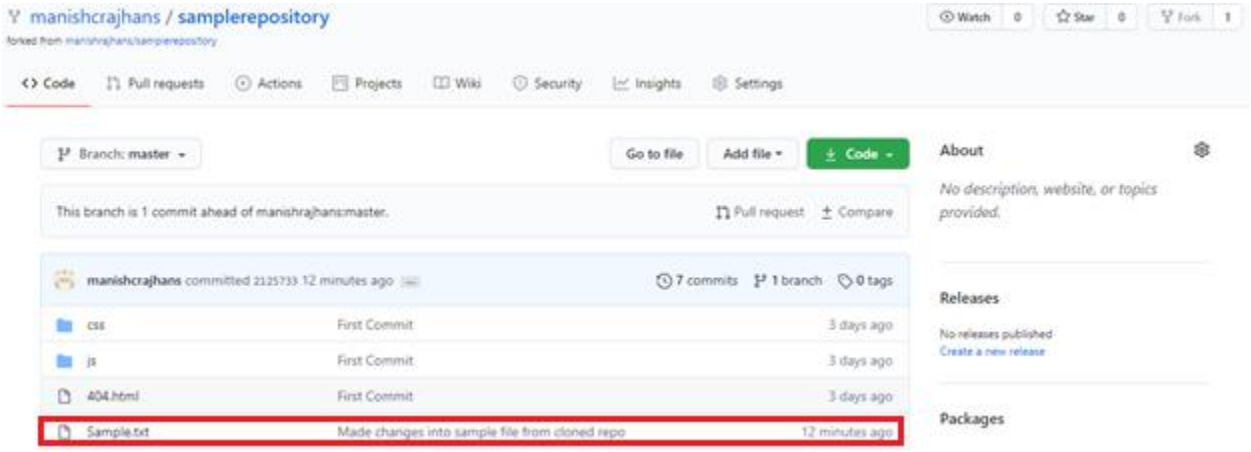
```
$ git add .
```

```
$ git commit -m "Made changes into sample file from cloned repo"
[master 2125733] Made changes into sample file from cloned repo
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Step13. Now push the changes to the Github repository.

```
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 356 bytes | 356.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/manishcrajhans/samplerepository.git
   687aaaf..2125733  master -> master
```

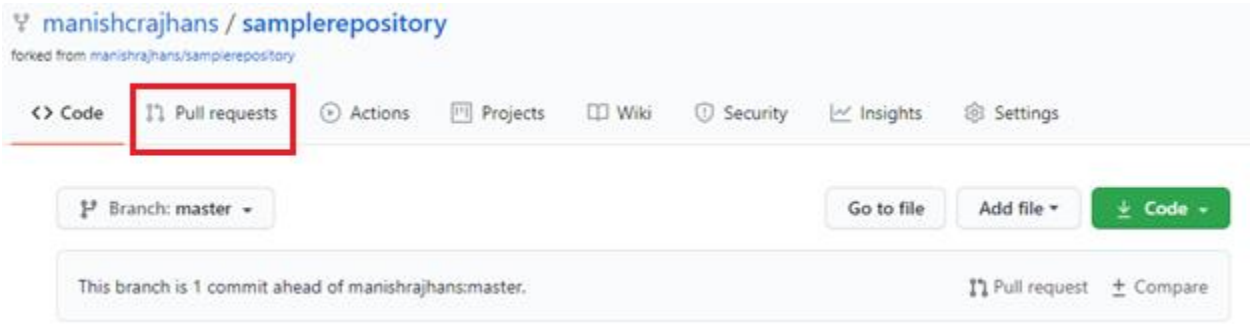
Step14. Now you can see the changes are added to the forked Github repository. Refresh GitHub repository, incase if you are unable to view the changes.



Remember this changes are done to the forked repository and not in the owners repository.

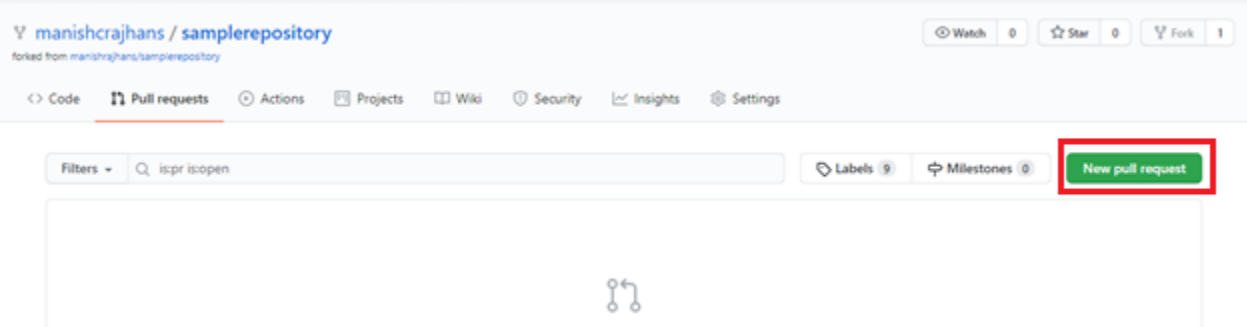
In order these changes to be accepted by the owner, you need to make pull request to the user.

Step15. Now in your forked repository, click on create Pull request.

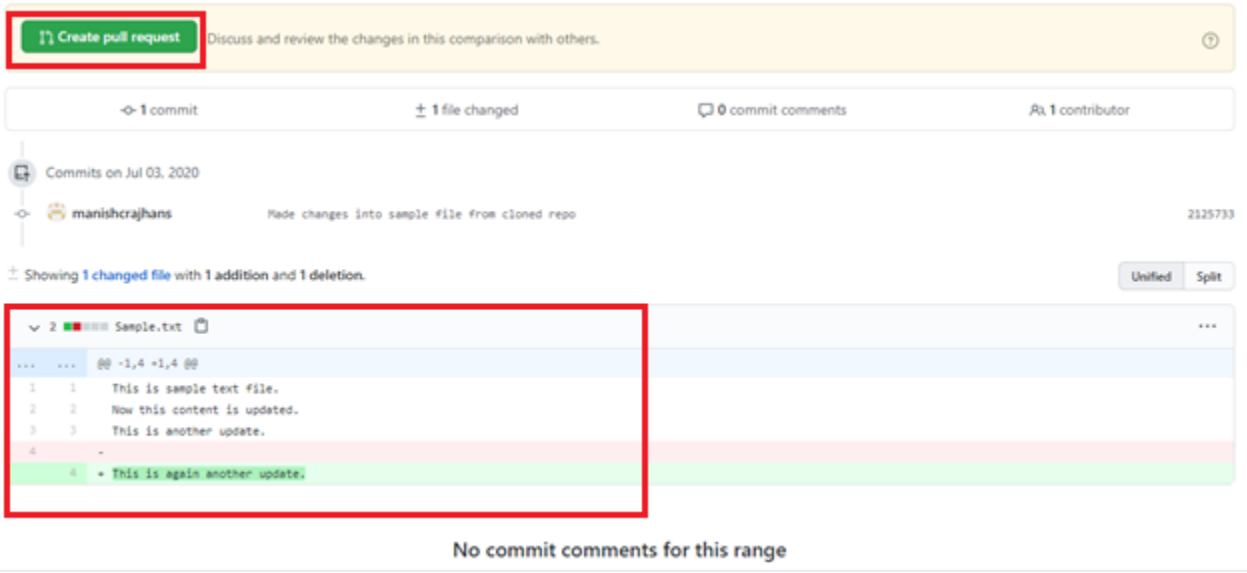


After clicking on this link, it will open the new page.

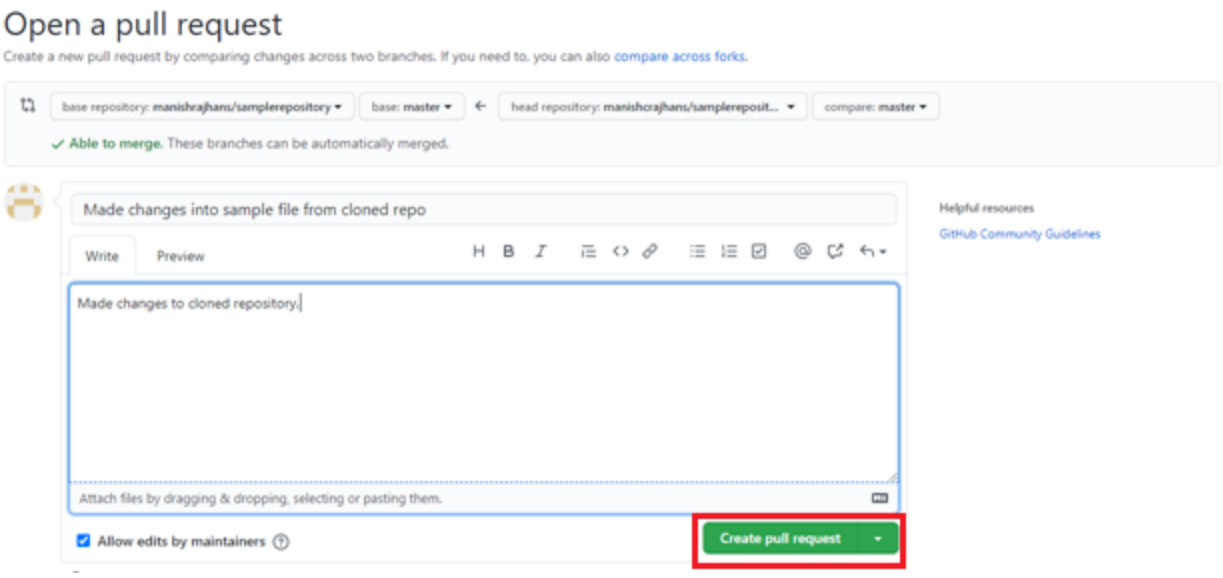
Step16. In the newly opened new page, click on New Pull Request.



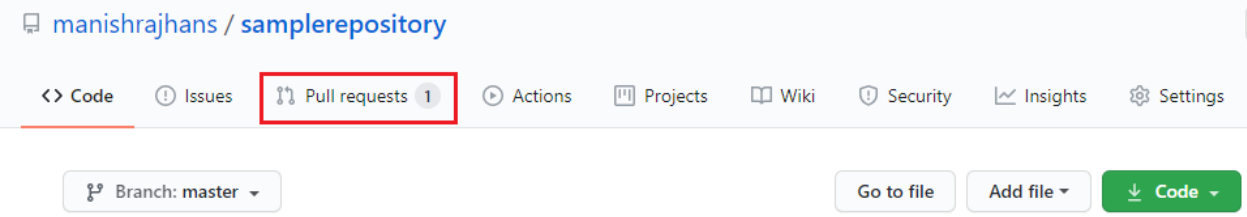
Step17. After clicking on New Pull Request, it will open a new page showing the changes done in the local repository, here click on Create Pull Request.



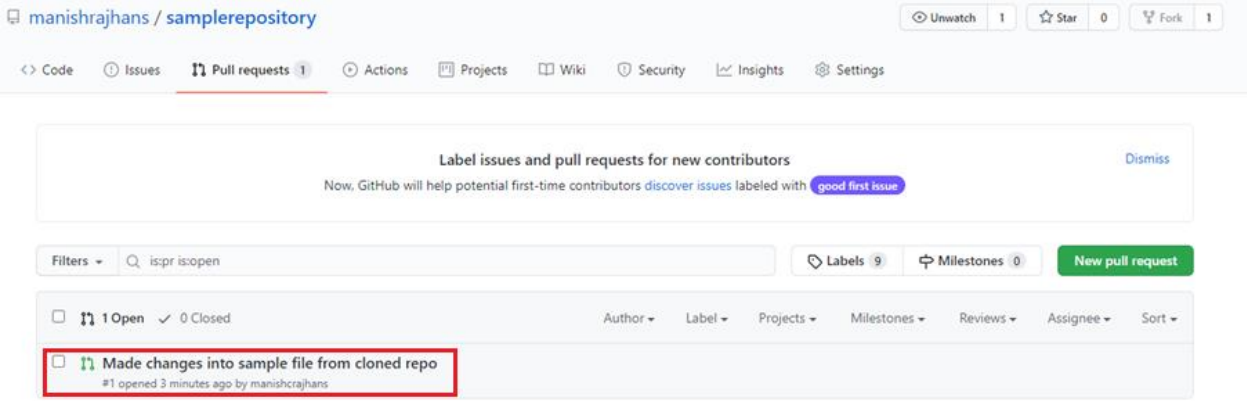
Step18. After clicking on Create Pull request, it will open a pull request and you can write the changes and then click on Pull Request.



Step19. Now you can see that the pull request is created by the forked repository user.

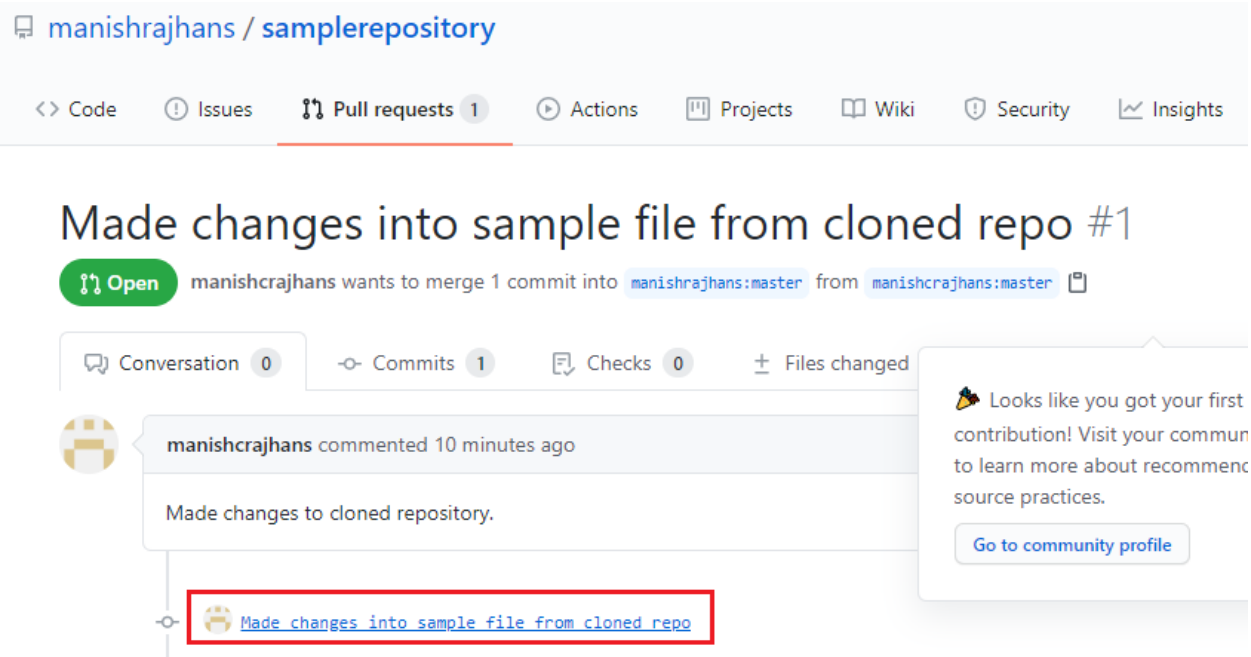


Step20. Once the request is initiated from the forked repository, owner of the repository will receive the notification in the pull request tab, if you (owner) can't see it, refresh the browser tab.



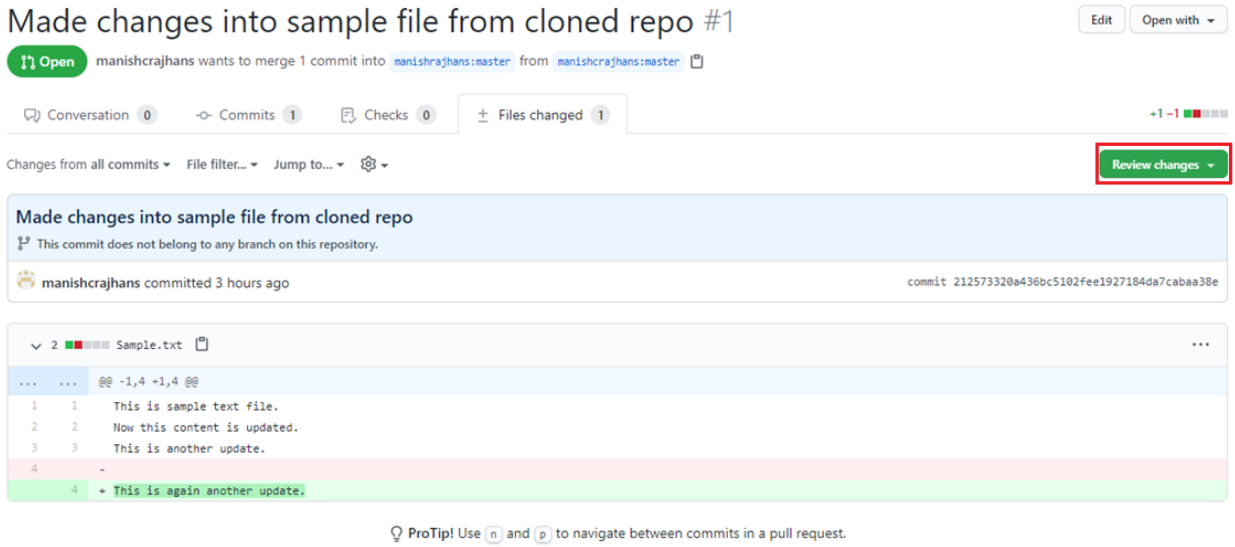
Now you can click on the received pull request to see the changes made.

Step21. Now after clicking on this, it will open the new page to show the changes made by the forked user into the repository.



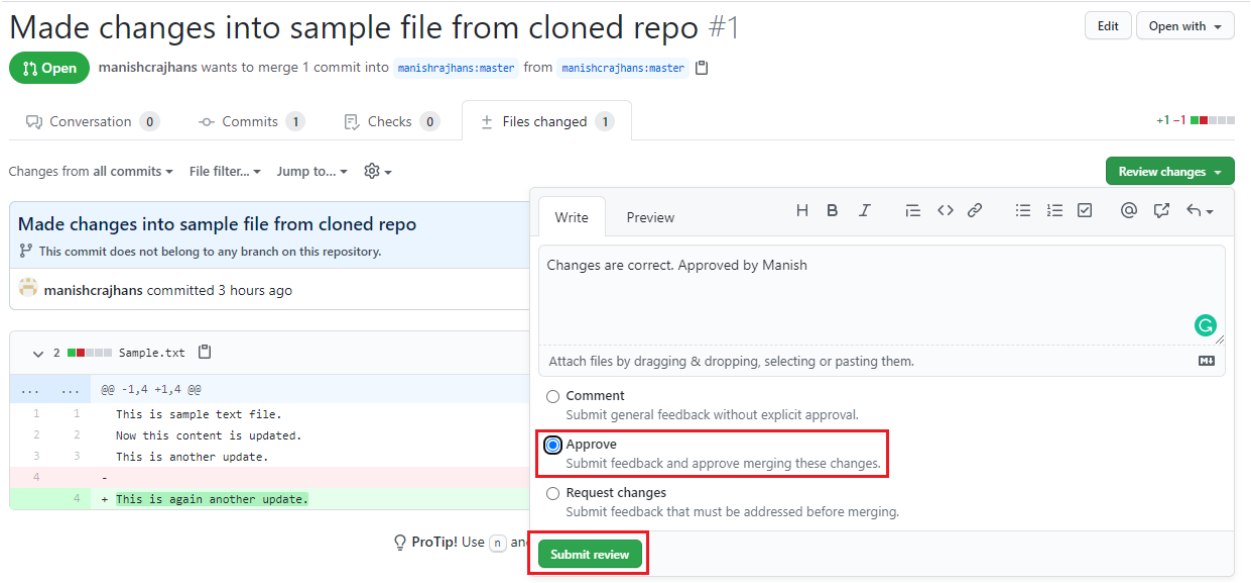
Click on the commit message to open the changes.

Step22. After you click on the message, it will open new page showing changes made with color codes showing what added, what deleted and what modified.

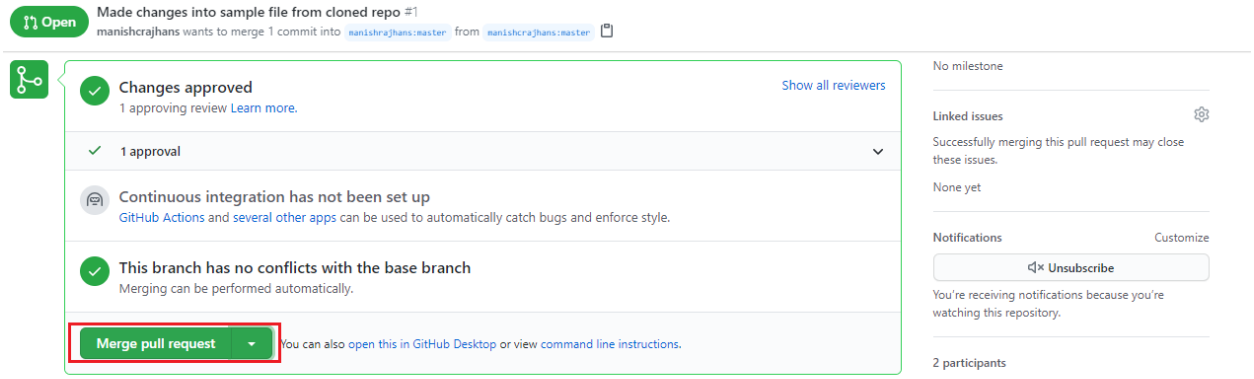


After having a look at the changes, now click on Review changes button.

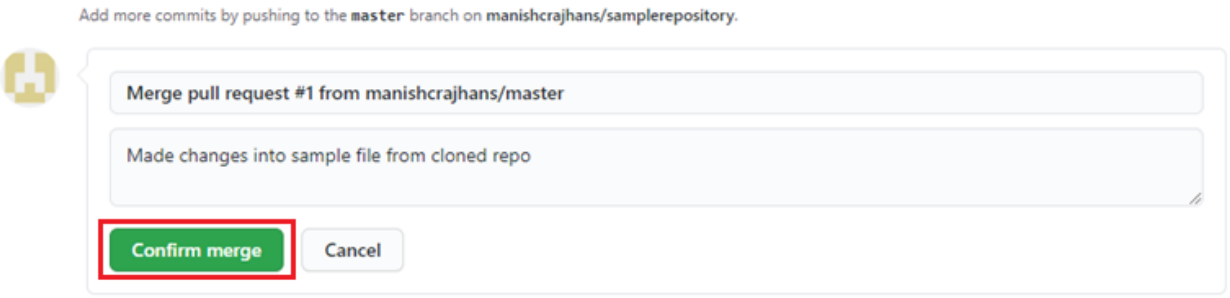
Step23. After clicking on the review changes button, you can approve, reject or request changes. As a part of this example, we are going to approve this changes and click on the Submit button.



Step24. After you clicked on Approve button, you may want to merge these changes to the owners branch. To do that click on Merge Pull request.



Step25. Finally now click on Confirm Merge



Step26. Now you can navigate to the owners repository and observe that changes committed to sample.txt is now available to the owner repository.

-: Branching the local repository:-

Step1. Navigate to the samplerepository's master branch in GitBash (if not already there), to check the available branches of repository type command.

```
$ git branch
* master
```

Step2. Now we will create a new feature branch, to do that use git branch command.

```
$ git branch NewFeature
```

Step2. Now switch to NewFeature branch with git checkout command.

```
mrajhans@LIN20001179 MINGW64 /c/study/gitrepos/SampleRepository (master)
$ git checkout NewFeature
Switched to branch 'NewFeature'

mrajhans@LIN20001179 MINGW64 /c/study/gitrepos/SampleRepository (NewFeature)
$
```

Step3. Check the status of this new branch.

```
$ git status
On branch NewFeature
nothing to commit, working tree clean
```

Step4. In the NewFeature branch modify the sample.txt file.

```
$ nano sample.txt
```

Step5. Check the status of the branch, it will show that file is modified.

```
mrajhans@LIN20001179 MINGW64 /c/study/gitrepos/SampleRepository (NewFeature)
$ git status
On branch NewFeature
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:   Sample.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

Step6. Now add the modified file into the staging area.

```
$ git add .
```

Step7. Commit the changes to the local repository.

```
$ git commit -m "First commit to NewFeaturesBranch"
[NewFeature 0e063e1] First commit to NewFeaturesBranch
 1 file changed, 2 insertions(+)
```

Step8. Now to merge the changes from the NewFeature branch to the master branch, checkout to the master branch.

```
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

Step9. After you are back on the master branch, now before merge, check the difference in the content with git diff command.

```
$ git diff NewFeature
diff --git a/Sample.txt b/Sample.txt
index f0e8c74..54a3e09 100644
--- a/Sample.txt
+++ b/Sample.txt
@@ -1,3 +1 @@
 This is sample text file.
-
-This new feature change added through feature branch.
```

Step10. If changes found to be appropriate, then you can merge the changes with the git merge command.

```
$ git merge NewFeature
Updating b78d1a7..0e063e1
Fast-forward
 Sample.txt | 2 ++
 1 file changed, 2 insertions(+)
```

Step11. After making changes to the master branch, still there exists NewFeatures branch. You can check this with git branch command.

```
$ git branch
  NewFeature
* master
```

Step12. If you feel there is no more need of NewFeature branch, then you may delete it as shown below.

```
$ git branch -d NewFeature
Deleted branch NewFeature (was 0e063e1).
```

Step13. Check the branches within the repository as below.

```
$ git branch
* master
```

-: Branching the local repository:-

You can check the history of write with git log command.