Assignment 1 for DevOps(CSIZG514)

October 2019

Instructor : Ms. Sonika Rathi

Submitted by:

Name: Vidhi Sethi

WILP email: 2019HT66115@wilp.bits-pilani.ac.in

Question:

GitHub Assignment:

ABC Organization would like to opt for the distributed version control system to upgrade their environment; where Git has been selected as the solution.

You have been assigned as a consultant to educate the migration process to move their Source Code from Centralized to Distributed systems. As a phase one, you would like to go ahead with a workshop to demonstrate the below operation to make the ABC team comfortable.

i) Create a Repository

ii) Add Two Directory and some raw code files to the repository

iii) Move Code from One directory to Another Directory

iv) Update one source code file and display the difference

v) Create a Branch

vi) Add some raw code to the branch

vii) Merge the Branch with Main line

And at the end provide the Summary of advantages of moving from Centralized Source Code to Distributed Version Control.

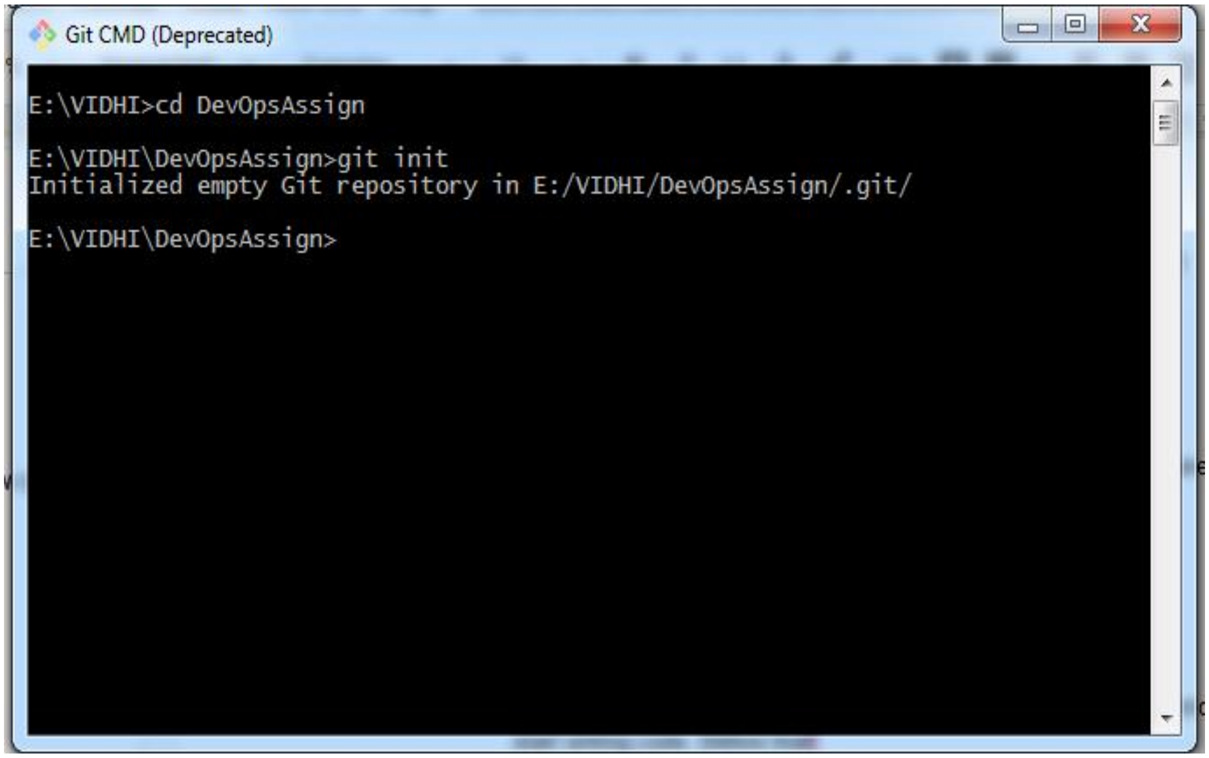
Note: Kindly share the screenshot's of the operations performed for above assignments. Make sure the screenshots contain your userID / LoginID to Identify it is not a copy paste content.

INDEX

i) Create a Repository:

a. Creating a repository in the local machine.

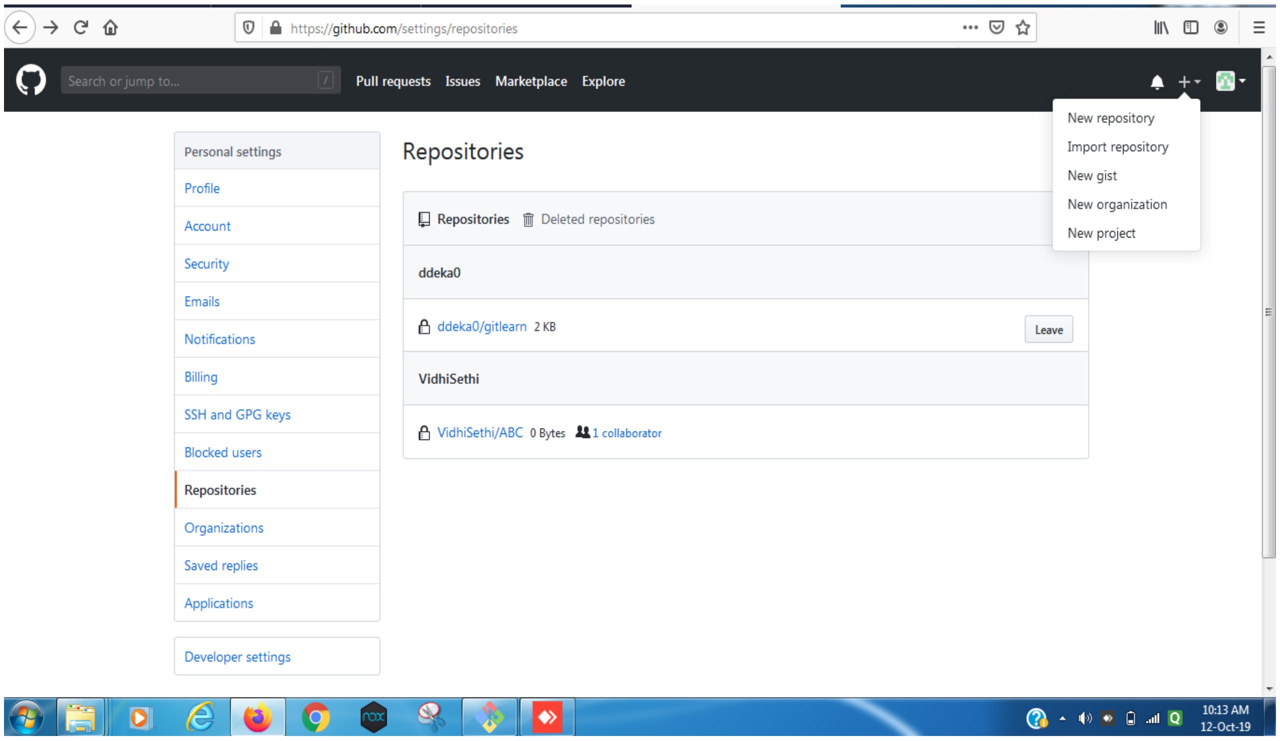
- `cd <dir_name>` (for example dir_name = DevOpsAssign)
- `git init`



```
E:\VIDHI>cd DevOpsAssign

E:\VIDHI\DevOpsAssign>git init
Initialized empty Git repository in E:/VIDHI/DevOpsAssign/.git/

E:\VIDHI\DevOpsAssign>
```

b. Creating a repository in Github

- Use GUI interface to create a repo named "DevOpsAssign"



https://github.com/settings/repositories

Search or jump to... Pull requests Issues Marketplace Explore

New repository
Import repository
New gist
New organization
New project

Personal settings
Profile
Account
Security
Emails
Notifications
Billing
SSH and GPG keys
Blocked users
Repositories
Organizations
Saved replies
Applications

Developer settings

Repositories

Repositories | Deleted repositories

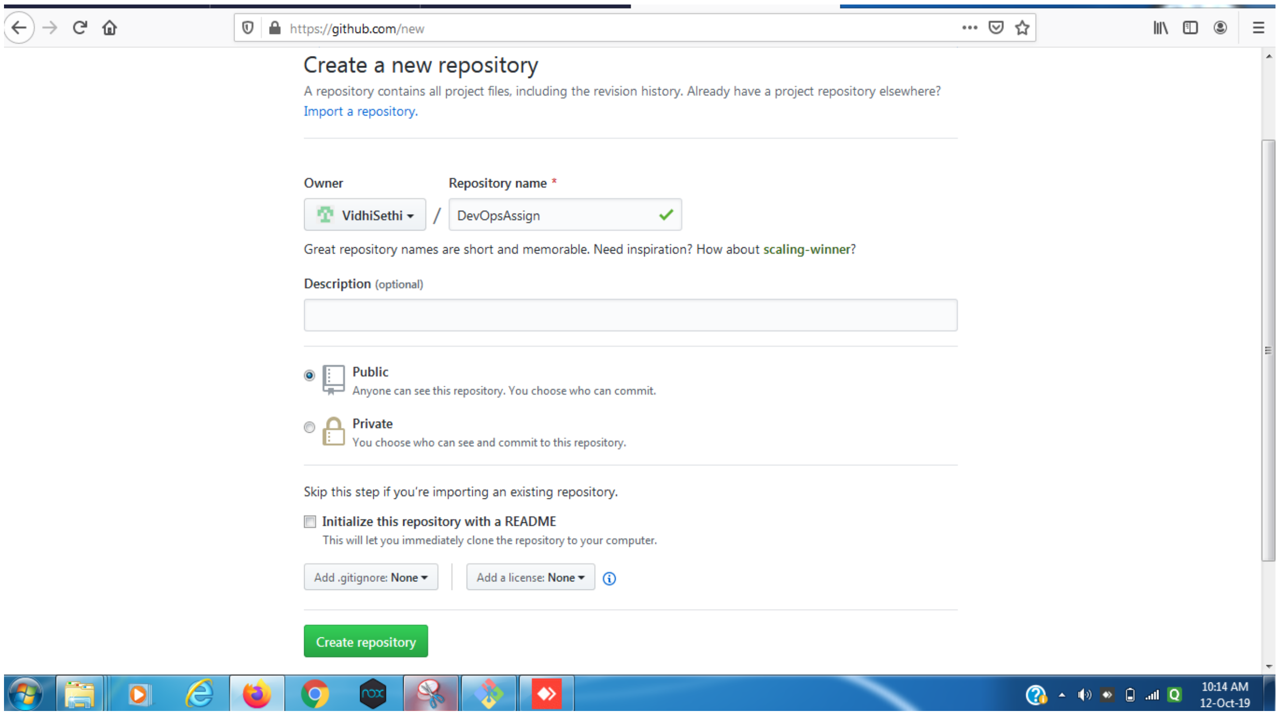
ddeka0

ddeka0/gitlearn 2 KB Leave

VidhiSethi

VidhiSethi/ABC 0 Bytes 1 collaborator

10:13 AM 12-Oct-19



https://github.com/new

Create a new repository

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

Owner: VidhiSethi / Repository name *: DevOpsAssign

Great repository names are short and memorable. Need inspiration? How about scaling-winner?

Description (optional)

(●) Public
Anyone can see this repository. You choose who can commit.

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

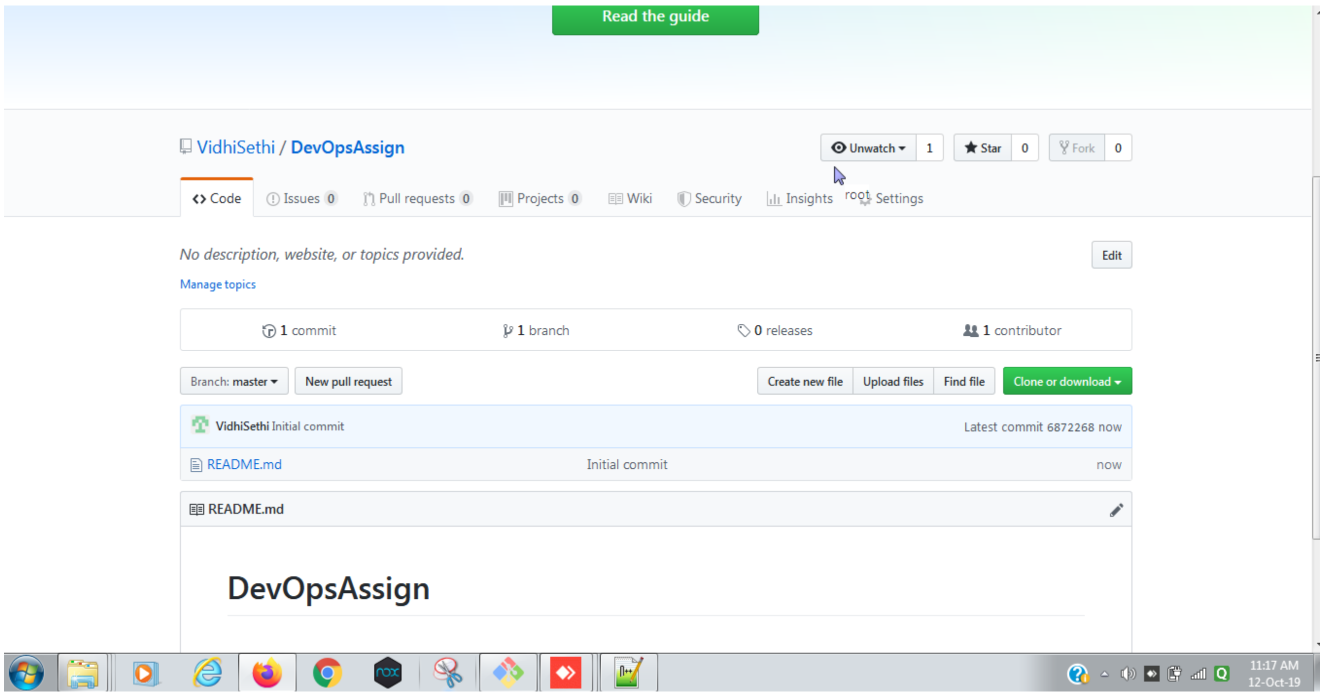
Skip this step if you're importing an existing repository.

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

Add .gitignore: None | Add a license: None

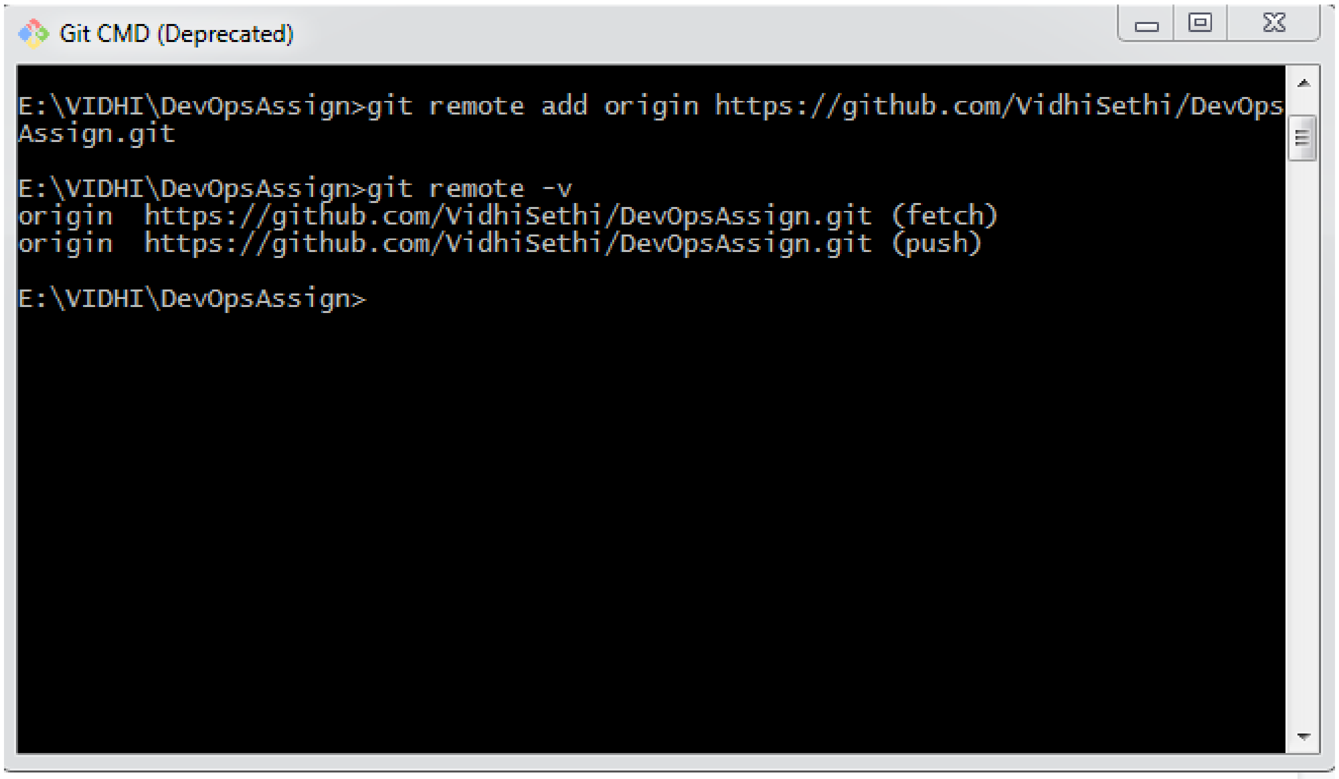
Create repository

10:14 AM 12-Oct-19



We need to add the remote repository to local git instance so that push and pull can be performed.

```
git remote add origin <git url>
```



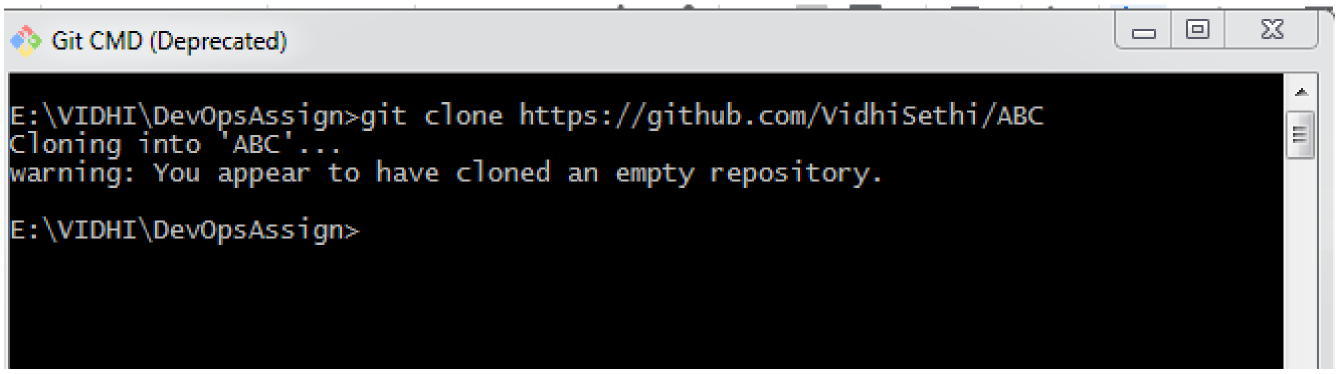
```
E:\VIDHI\DevOpsAssign>git remote add origin https://github.com/VidhiSethi/DevOps
Assign.git

E:\VIDHI\DevOpsAssign>git remote -v
origin  https://github.com/VidhiSethi/DevOpsAssign.git (fetch)
origin  https://github.com/VidhiSethi/DevOpsAssign.git (push)

E:\VIDHI\DevOpsAssign>
```

Note: In order to avoid typing user name and password in git bash. Generate the asymmetric key and save it in the local system.

c. Instead of converting an existing local directory to a git repository we can also clone an existing remote repository to our local machine.

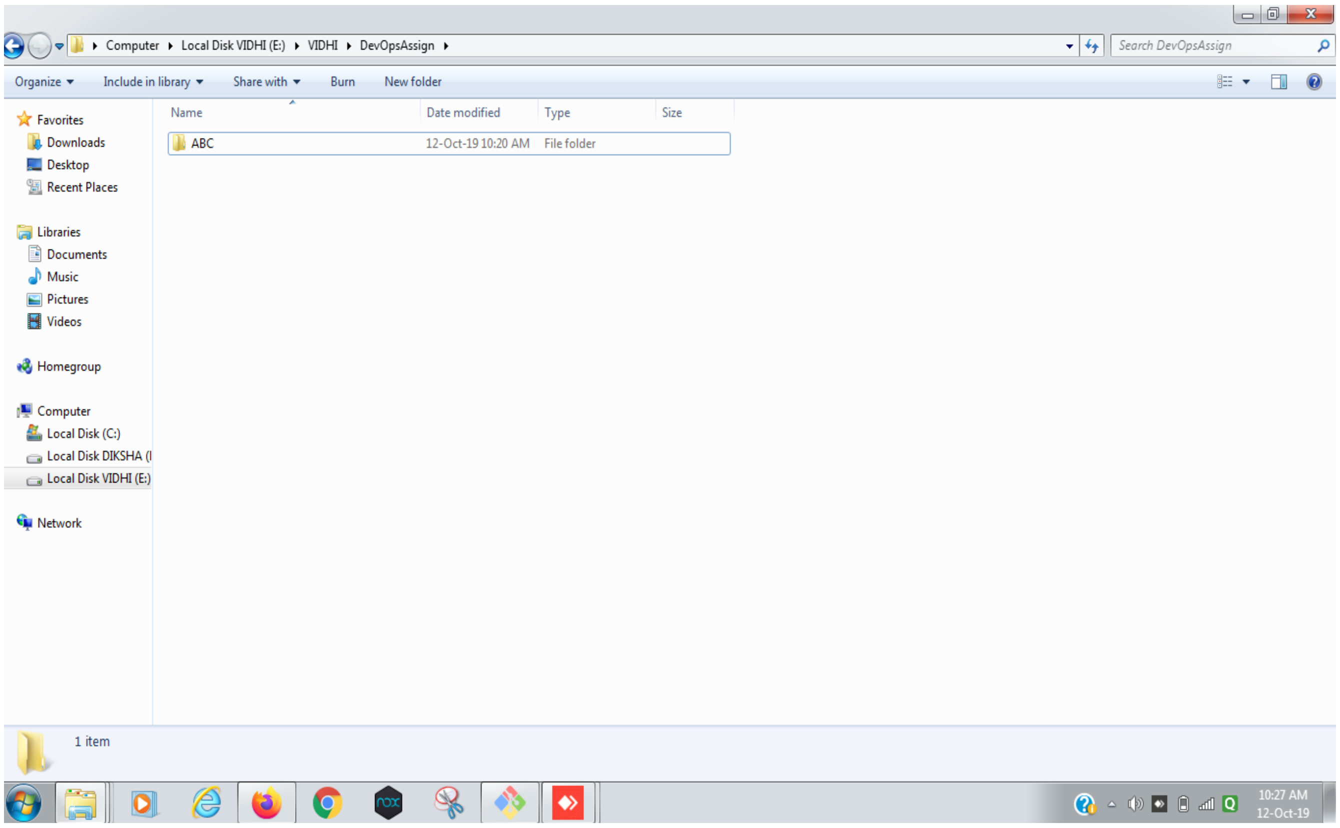
```
git clone https://github.com/VidhiSethi/ABC
```

Here, 'ABC' is existing remote repository.



Git CMD (Deprecated)

```
E:\VIDHI\DevOpsAssign>git clone https://github.com/VidhiSethi/ABC
Cloning into 'ABC'...
warning: You appear to have cloned an empty repository.

E:\VIDHI\DevOpsAssign>
```



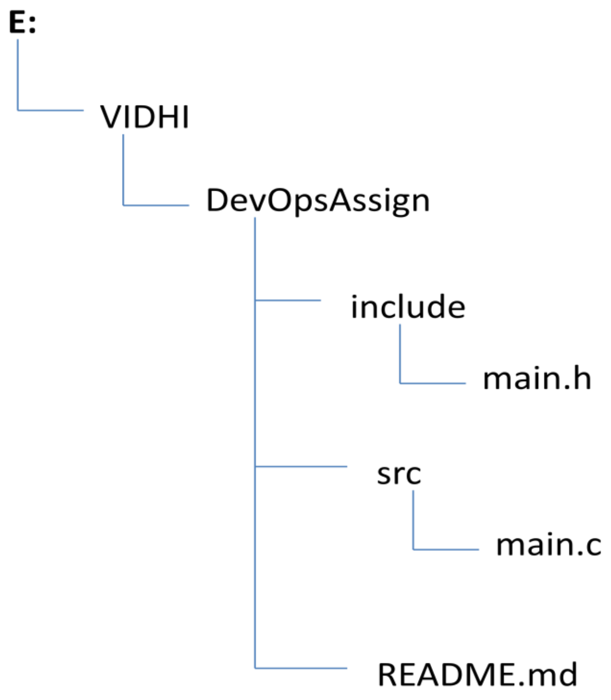
Name	Date modified	Type	Size
ABC	12-Oct-19 10:20 AM	File folder	

ii) Add Two Directory and some raw code files to the repository

Once ‘ABC’ organization is ready with their local repository and remote repository, they can start writing code.

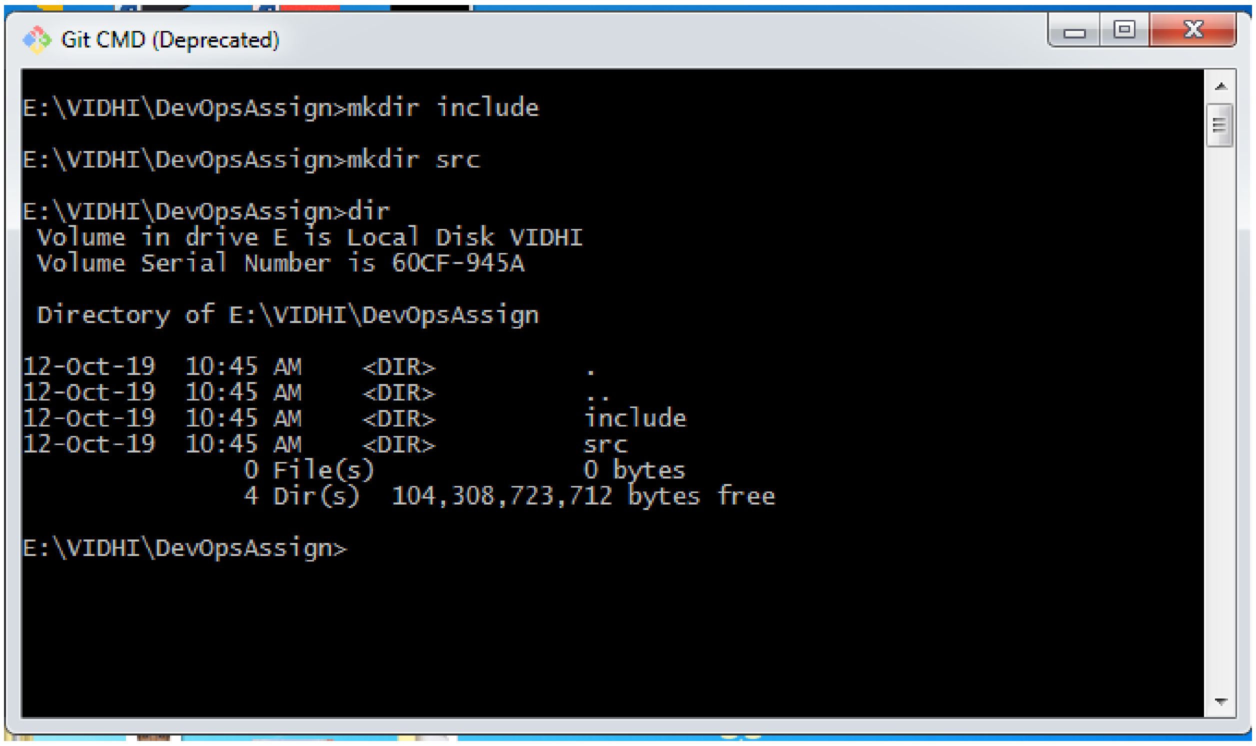
Let’s assume they are about to work on a C project. Before adding any code, they need to have a project structure. This project structure could be as shown below.

It consists of two directory **‘include’** and **‘src’** and two files **‘main.h’** and **‘main.c’.**

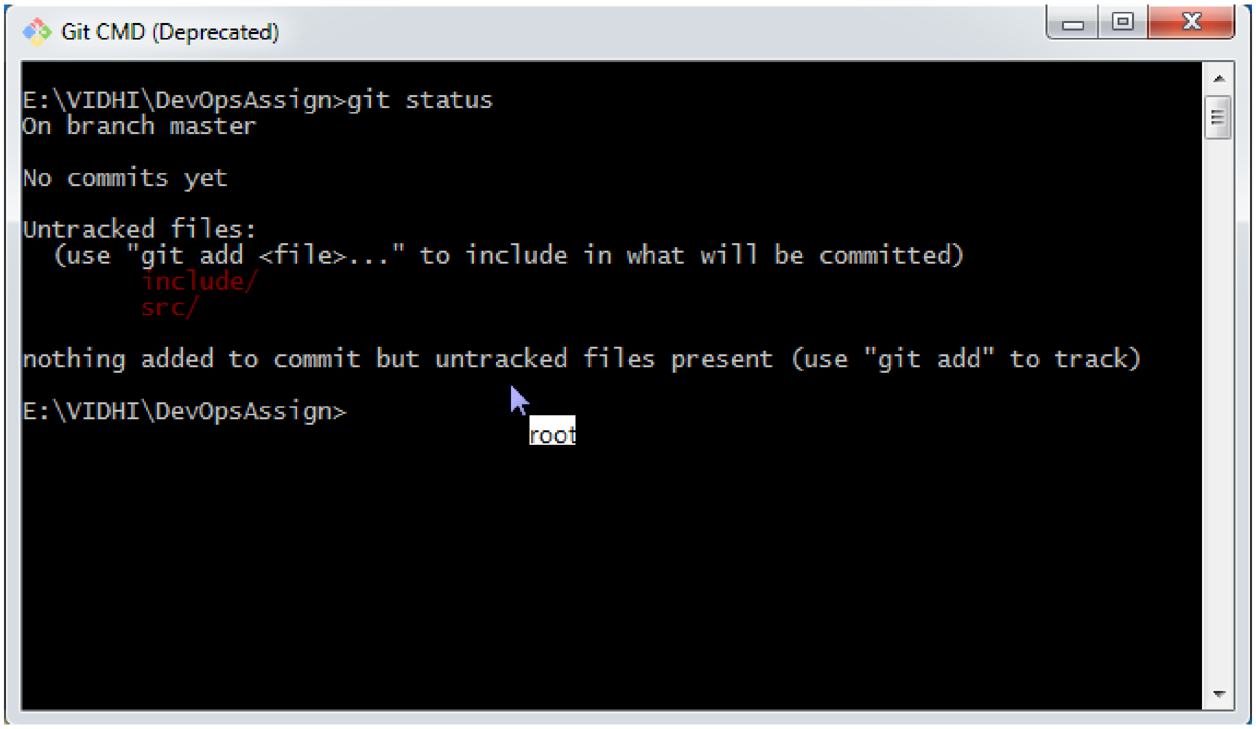


When we create two empty directories and try to do `git status` then git will not show any unstaged changes. After we added few source files, git shows unstaged (or untracked) changes in `git status` report as shown in the following figures.

```
mkdir <directory name>
```


Git CMD (Deprecated)

```
E:\VIDHI\DevOpsAssign>mkdir include

E:\VIDHI\DevOpsAssign>mkdir src

E:\VIDHI\DevOpsAssign>dir
 Volume in drive E is Local Disk VIDHI
 Volume Serial Number is 60CF-945A

 Directory of E:\VIDHI\DevOpsAssign

12-Oct-19  10:45 AM    <DIR>          .
12-Oct-19  10:45 AM    <DIR>          ..
12-Oct-19  10:45 AM    <DIR>          include
12-Oct-19  10:45 AM    <DIR>          src
               0 File(s)              0 bytes
               4 Dir(s)  104,308,723,712 bytes free

E:\VIDHI\DevOpsAssign>
```

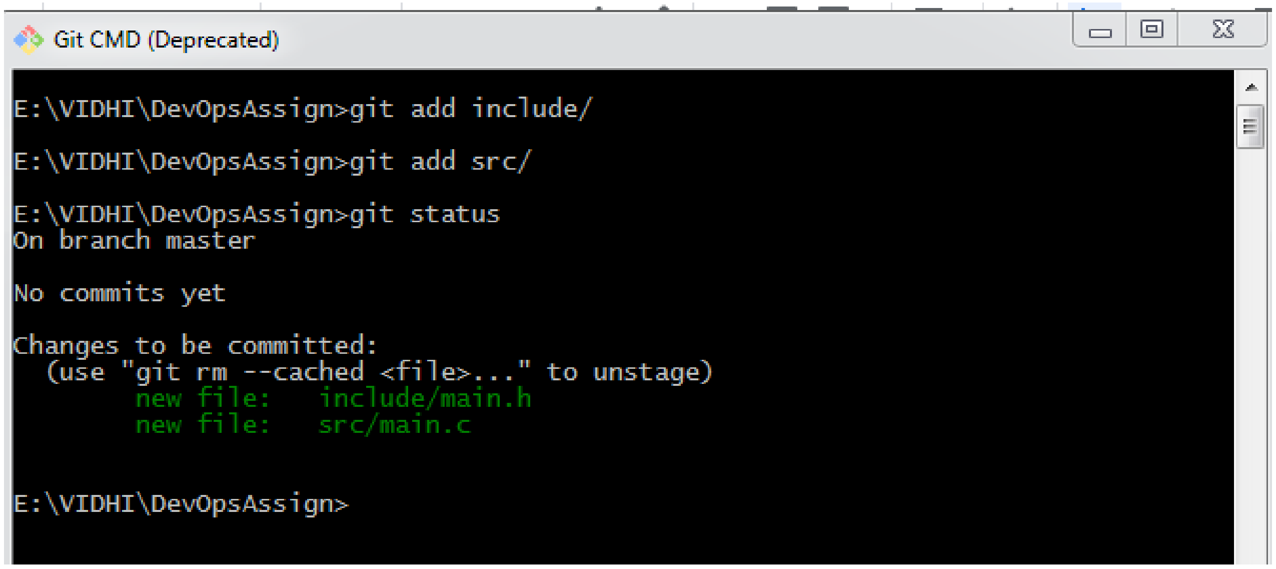

Git CMD (Deprecated)

```
E:\VIDHI\DevOpsAssign>git status
On branch master

No commits yet

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

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

E:\VIDHI\DevOpsAssign>
```

After, these operations, we need to add the files to the git staging area so that git can track the changes. Git will create objects inside .git directory so that it can track changes of the corresponding files.

a. `git add include/`

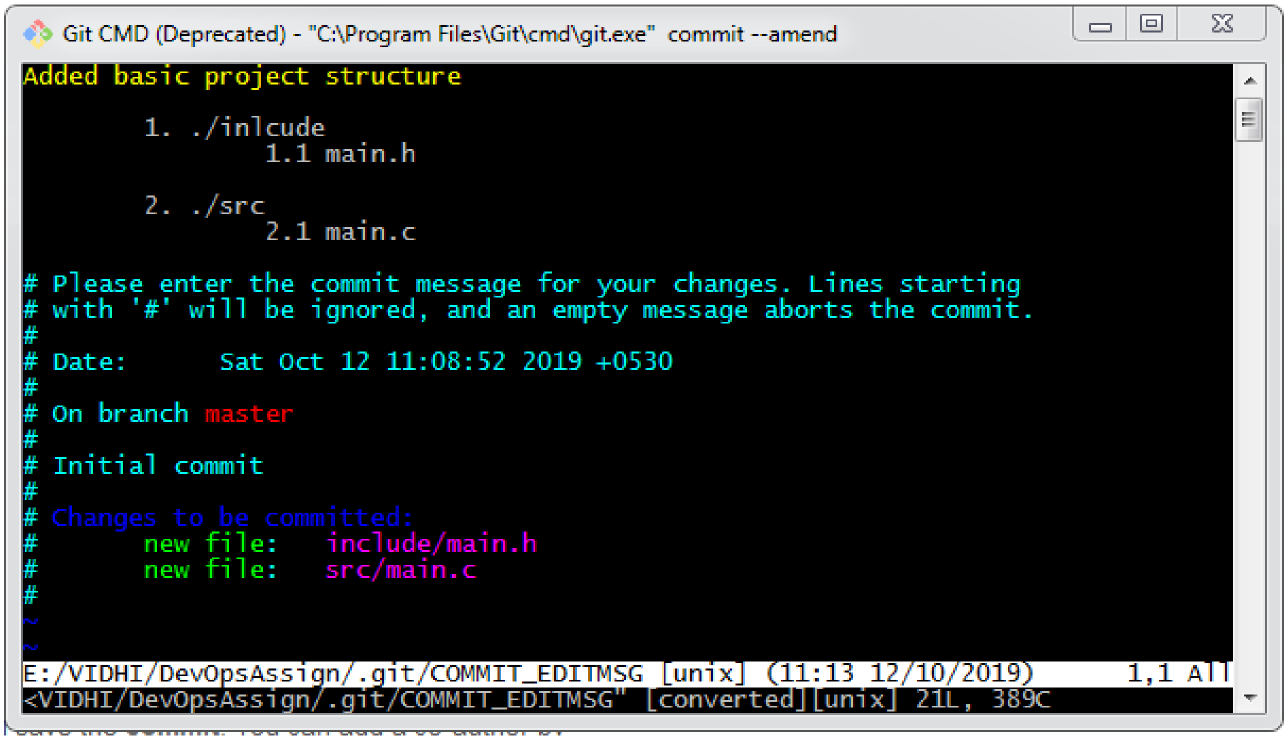
b. `git add src/`



Git CMD (Deprecated)

```
E:\VIDHI\DevOpsAssign>git add include/

E:\VIDHI\DevOpsAssign>git add src/

E:\VIDHI\DevOpsAssign>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   include/main.h
        new file:   src/main.c


E:\VIDHI\DevOpsAssign>
```

To commit the changes to the local repository: `git commit`

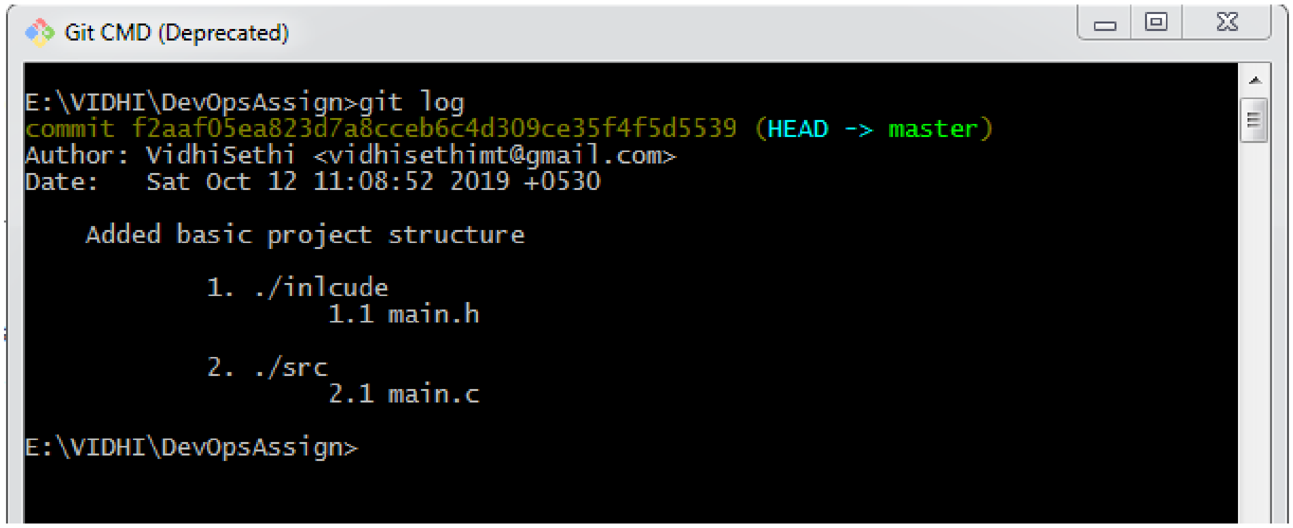


Git CMD (Deprecated) - "C:\Program Files\Git\cmd\git.exe" commit --amend

```
Added basic project structure

        1. ./inlcude
                1.1 main.h

        2. ./src
                2.1 main.c

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Sat Oct 12 11:08:52 2019 +0530
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#       new file:   include/main.h
#       new file:   src/main.c
#
~
~
E:/VIDHI/DevOpsAssign/.git/COMMIT_EDITMSG [unix] (11:13 12/10/2019)      1,1 All
<VIDHI/DevOpsAssign/.git/COMMIT_EDITMSG" [converted][unix] 21L, 389C
```

Check using :

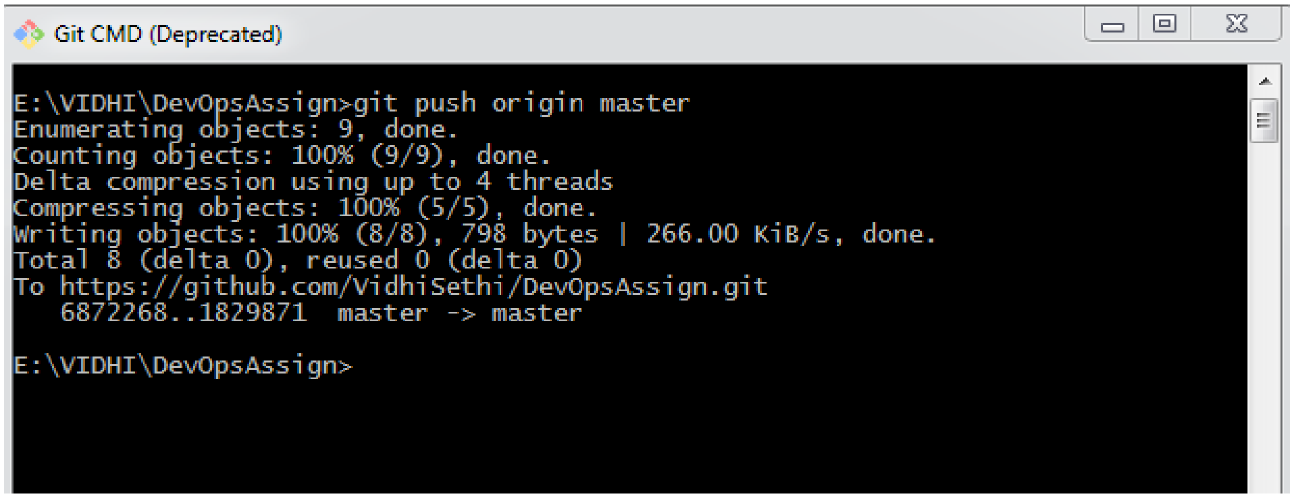
```
git log
```


Git CMD (Deprecated)

```
E:\VIDHI\DevOpsAssign>git log
commit f2aaf05ea823d7a8cceb6c4d309ce35f4f5d5539 (HEAD -> master)
Author: VidhiSethi <vidhisethimt@gmail.com>
Date:   Sat Oct 12 11:08:52 2019 +0530

    Added basic project structure

            1. ./inlcude
                    1.1 main.h

            2. ./src
                    2.1 main.c

E:\VIDHI\DevOpsAssign>
```

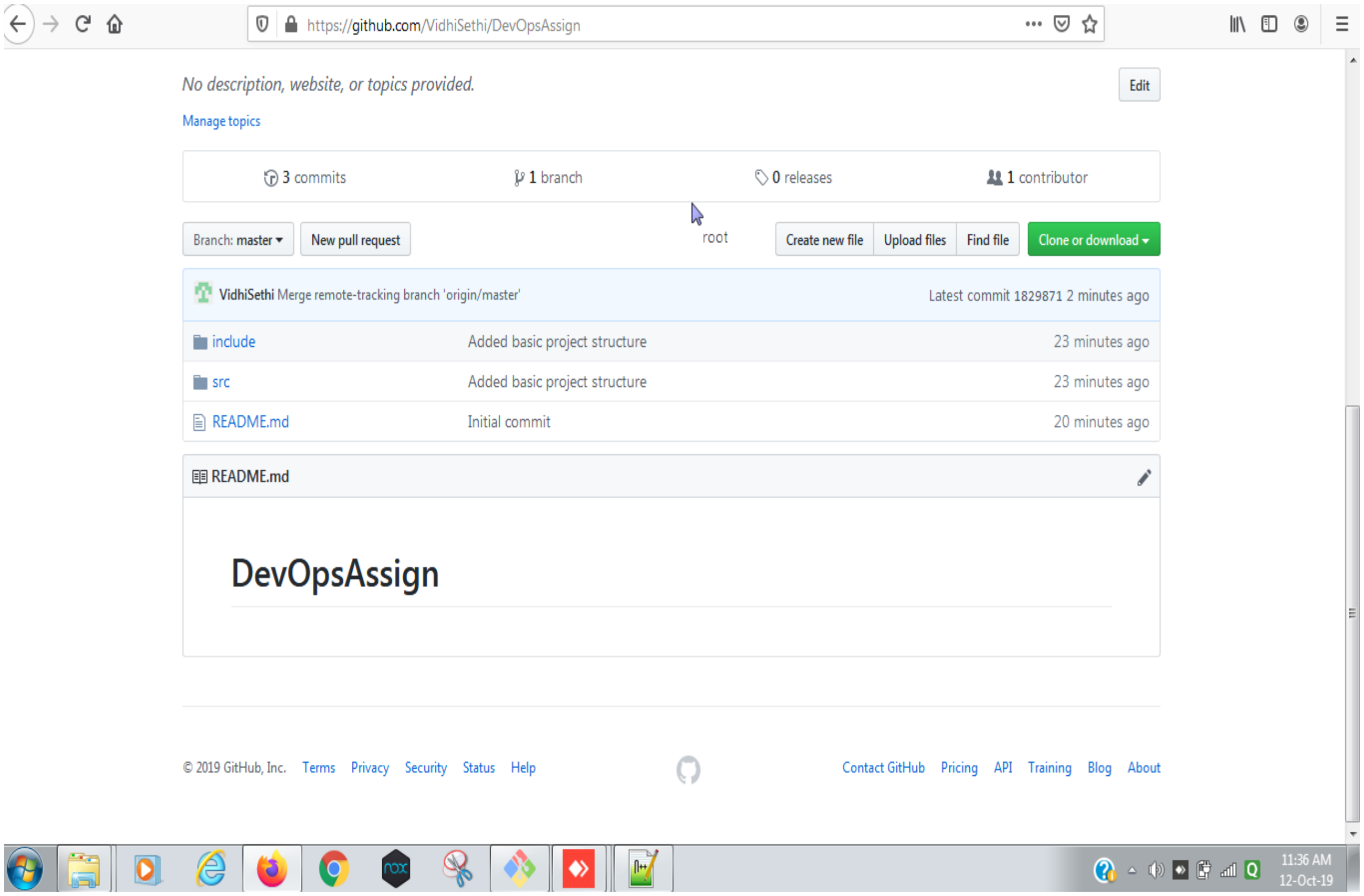
Copy changes to the remote repository using :

```
git push origin master
```


Git CMD (Deprecated)

```
E:\VIDHI\DevOpsAssign>git push origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (8/8), 798 bytes | 266.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0)
To https://github.com/VidhiSethi/DevOpsAssign.git
   6872268..1829871  master -> master

E:\VIDHI\DevOpsAssign>
```

Changes copied to the remote repository as shown in below picture:


https://github.com/VidhiSethi/DevOpsAssign

No description, website, or topics provided.

Manage topics

3 commits | 1 branch | 0 releases | 1 contributor

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

VidhiSethi Merge remote-tracking branch 'origin/master' — Latest commit 1829871 2 minutes ago

include	Added basic project structure	23 minutes ago
src	Added basic project structure	23 minutes ago
README.md	Initial commit	20 minutes ago

README.md

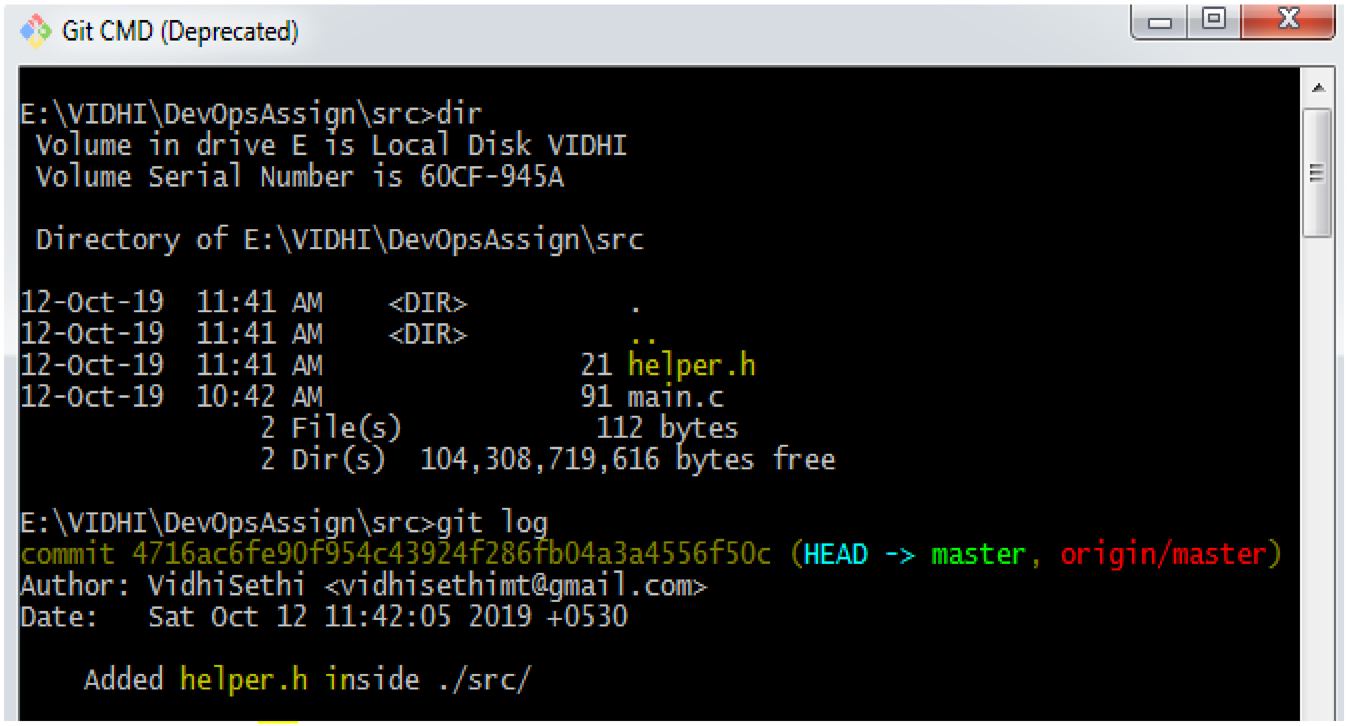
DevOpsAssign

© 2019 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

11:36 AM 12-Oct-19

iii) Move Code from One directory to Another Directory

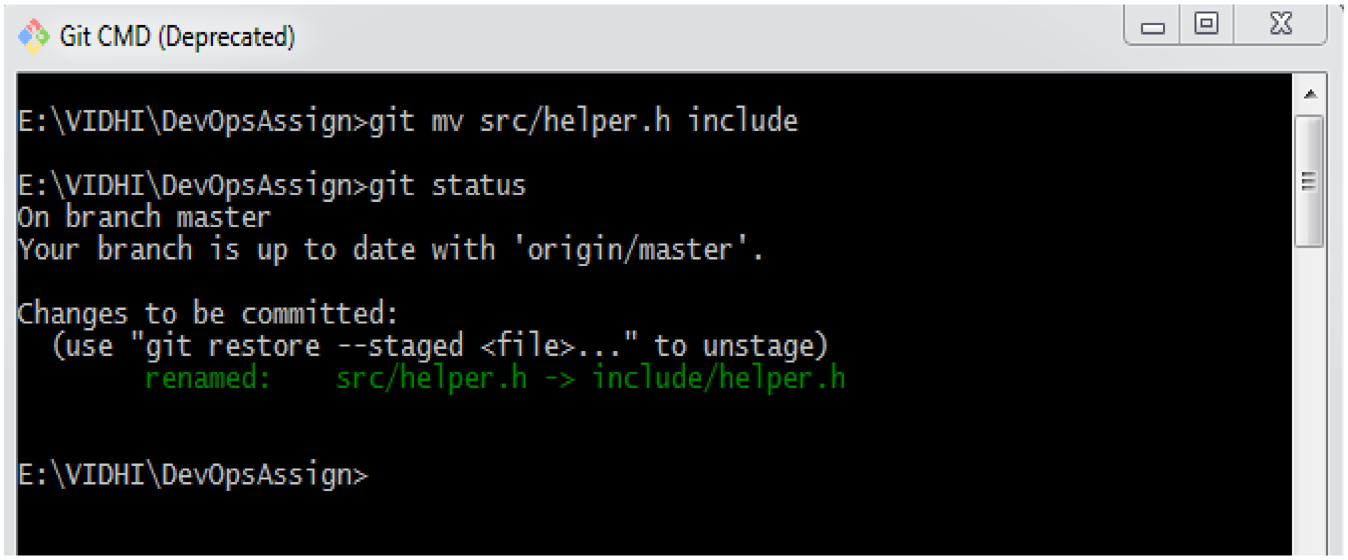
First, I have created ‘**helper.h**’ inside ‘**src**’ folder



```
E:\VIDHI\DevOpsAssign\src>dir
 Volume in drive E is Local Disk VIDHI
 Volume Serial Number is 60CF-945A

 Directory of E:\VIDHI\DevOpsAssign\src

12-Oct-19  11:41 AM    <DIR>          .
12-Oct-19  11:41 AM    <DIR>          ..
12-Oct-19  11:41 AM                21 helper.h
12-Oct-19  10:42 AM                91 main.c
               2 File(s)            112 bytes
               2 Dir(s)  104,308,719,616 bytes free

E:\VIDHI\DevOpsAssign\src>git log
commit 4716ac6fe90f954c43924f286fb04a3a4556f50c (HEAD -> master, origin/master)
Author: VidhiSethi <vidhisethimt@gmail.com>
Date:   Sat Oct 12 11:42:05 2019 +0530

    Added helper.h inside ./src/
```

Now, move ’ **helper.h** ‘ from **./src/** to **./include/ .**

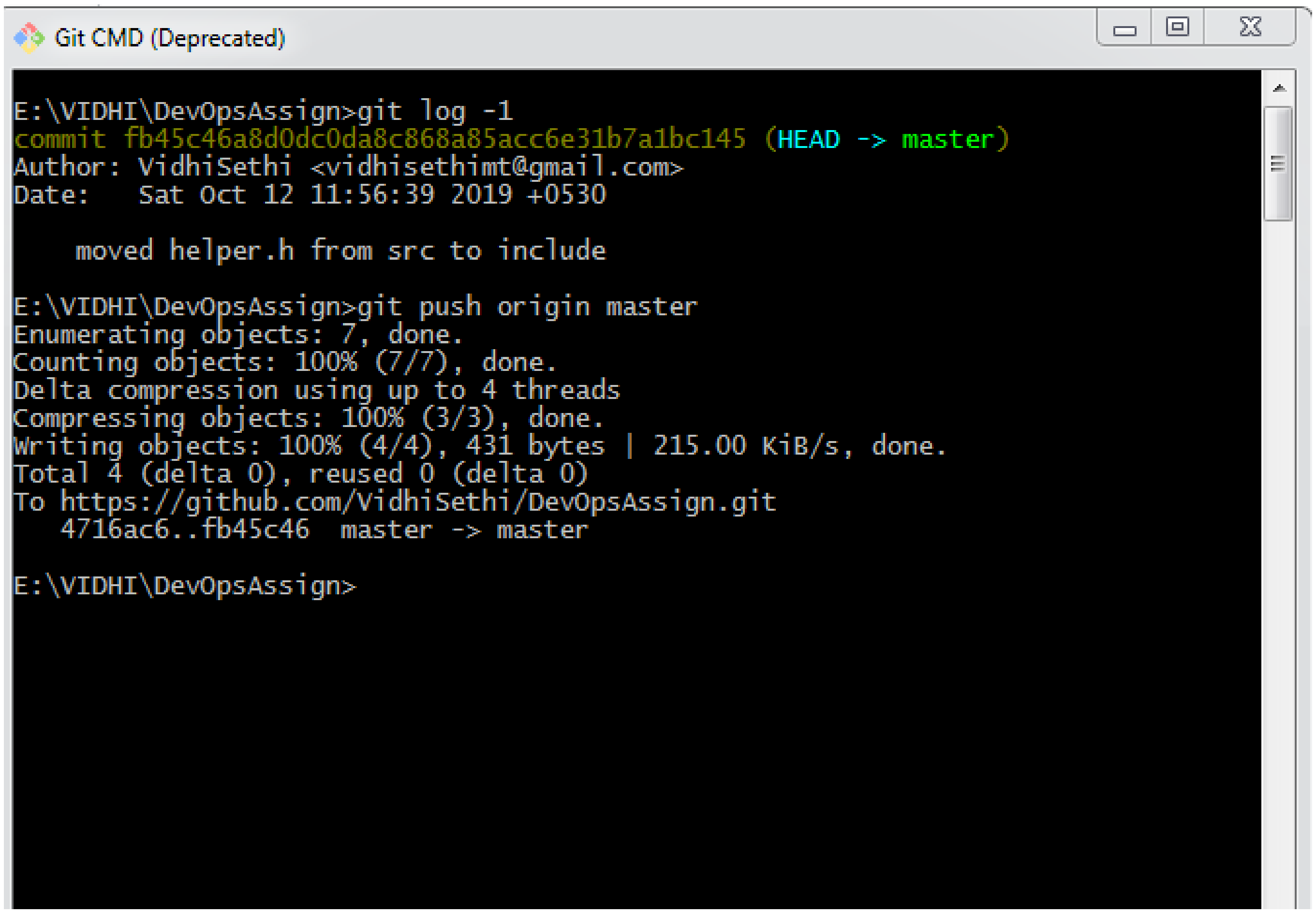
```
#git mv <file_from> <file_to>

git mv src/helper.h include
```



```
E:\VIDHI\DevOpsAssign>git mv src/helper.h include

E:\VIDHI\DevOpsAssign>git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    src/helper.h -> include/helper.h


E:\VIDHI\DevOpsAssign>
```

Check using :

```
git log
```



Git CMD (Deprecated)

```
E:\VIDHI\DevOpsAssign>git log -1
commit fb45c46a8d0dc0da8c868a85acc6e31b7a1bc145 (HEAD -> master)
Author: VidhiSethi <vidhisethimt@gmail.com>
Date:   Sat Oct 12 11:56:39 2019 +0530

    moved helper.h from src to include

E:\VIDHI\DevOpsAssign>git push origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 431 bytes | 215.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/VidhiSethi/DevOpsAssign.git
   4716ac6..fb45c46  master -> master

E:\VIDHI\DevOpsAssign>
```

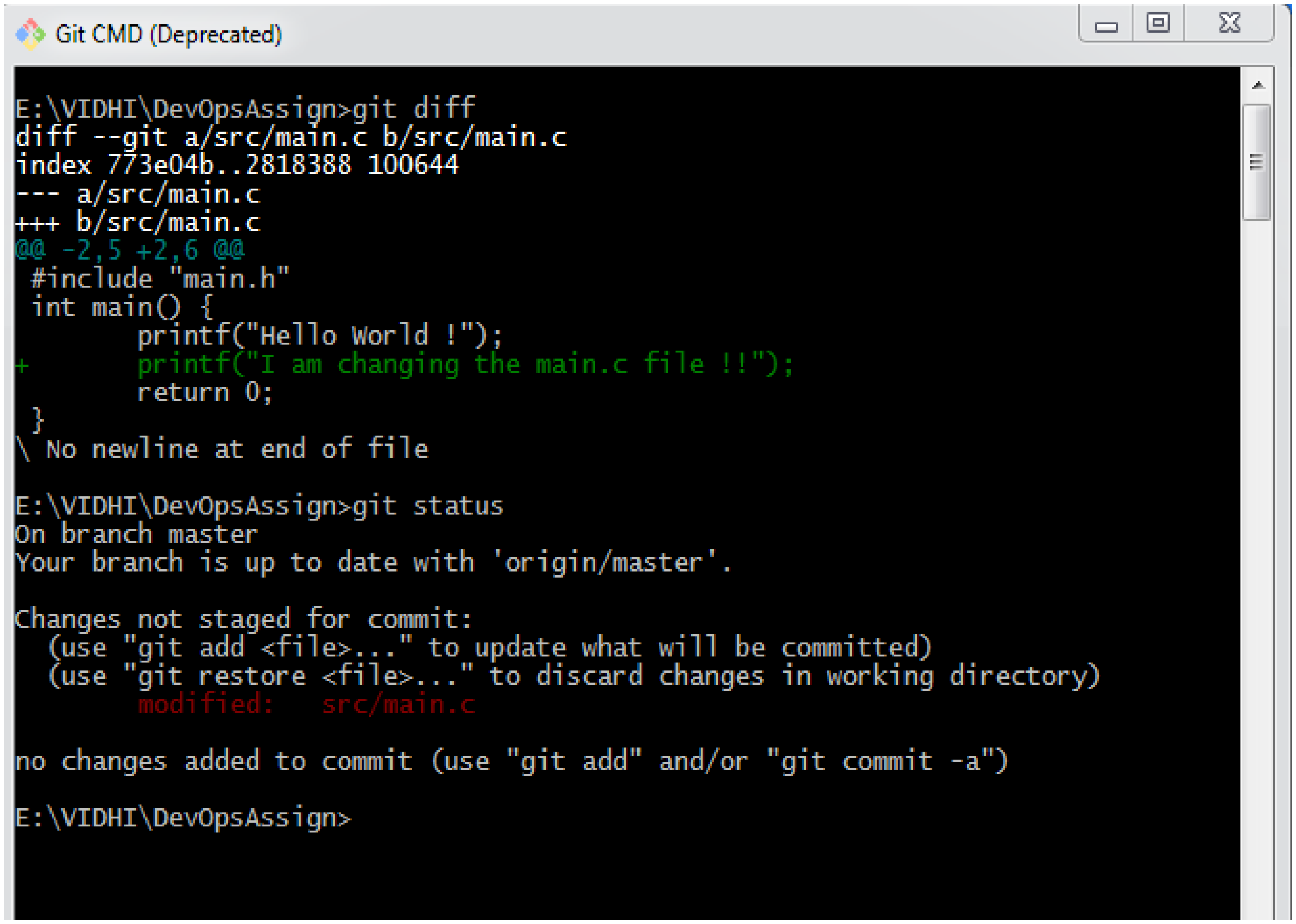
iv) Update one source code file and display the difference

I updated the **'main.c'** file in **'src'** folder and added the line :

```
printf("I am changing the main.c file !!");
```

To display the difference :

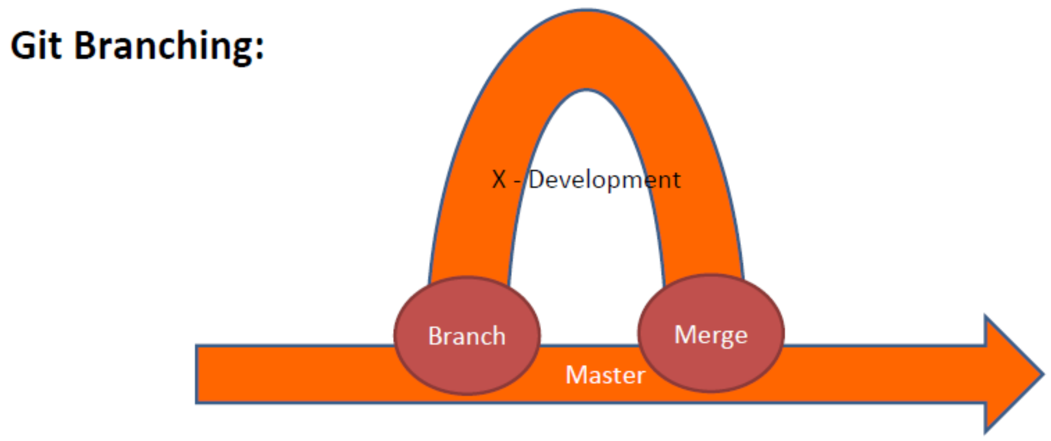
```
git diff
```



Git CMD (Deprecated)

```
E:\VIDHI\DevOpsAssign>git diff
diff --git a/src/main.c b/src/main.c
index 773e04b..2818388 100644
--- a/src/main.c
+++ b/src/main.c
@@ -2,5 +2,6 @@
 #include "main.h"
 int main() {
        printf("Hello World !");
+       printf("I am changing the main.c file !!");
        return 0;
 }
\ No newline at end of file

E:\VIDHI\DevOpsAssign>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:   src/main.c

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

E:\VIDHI\DevOpsAssign>
```

v) Create a Branch



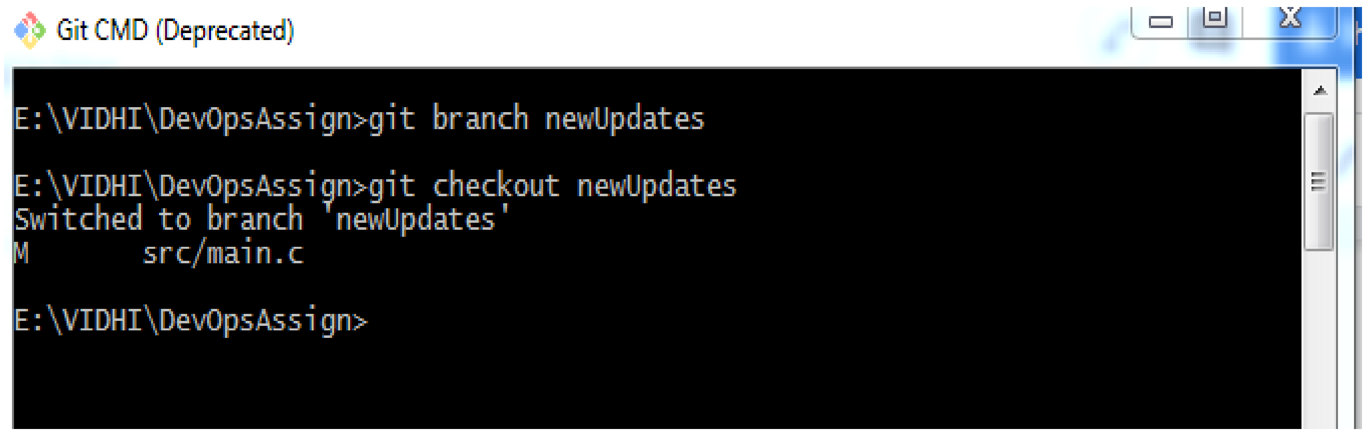
Create a new branch named **‘newUpdates’**.

```
#git branch <branch name>

git branch newUpdates
```

Git keeps a special pointer called HEAD.HEAD acts as a pointer to the local branch you’re currently on.The `git branch` command only created a new branch, it didn’t switch to that branch, you will be still on master branch.

To switch to an existing branch, you run :

```
#git checkout <branch name>

#git checkout newUpdates
```

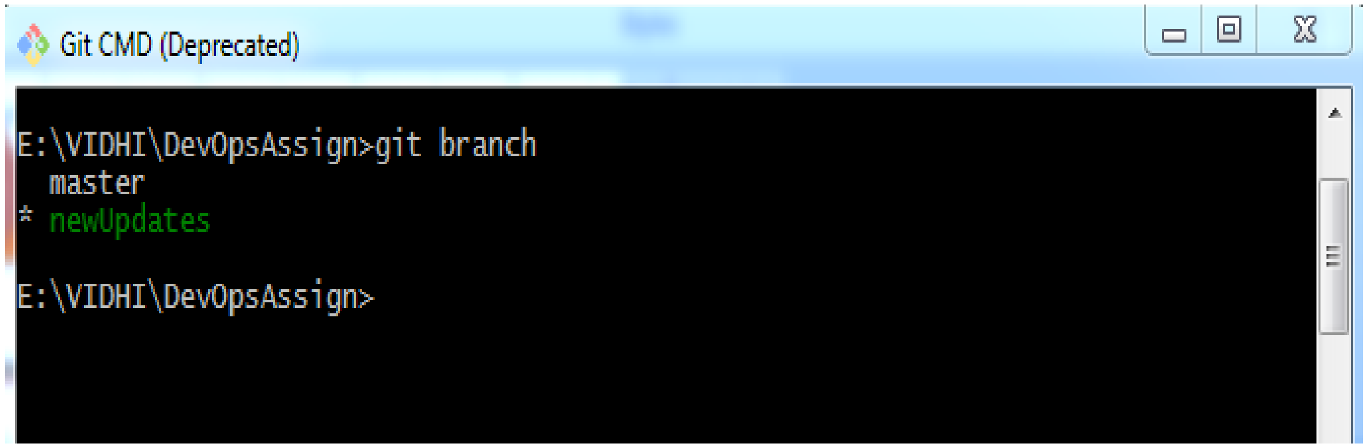


```
E:\VIDHI\DevOpsAssign>git branch newUpdates

E:\VIDHI\DevOpsAssign>git checkout newUpdates
Switched to branch 'newUpdates'
M       src/main.c

E:\VIDHI\DevOpsAssign>
```

Check for the created branches using :

```
git branch
```

Current branch is highlighted in green.

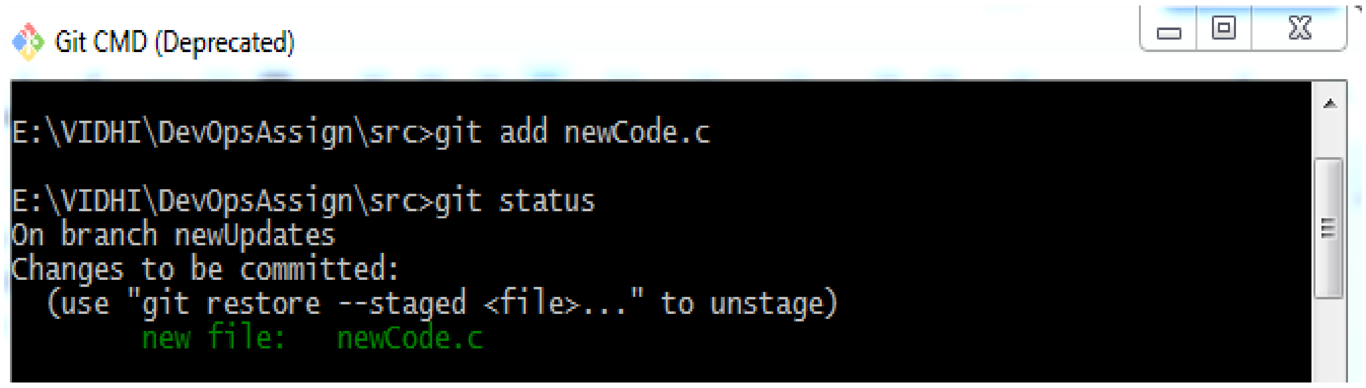


```
E:\VIDHI\DevOpsAssign>git branch
  master
* newUpdates

E:\VIDHI\DevOpsAssign>
```

vi) Add some raw code to the branch

I added the file **'newCode.c'** in the **'src'** folder while working in **'newUpdates'** branch.

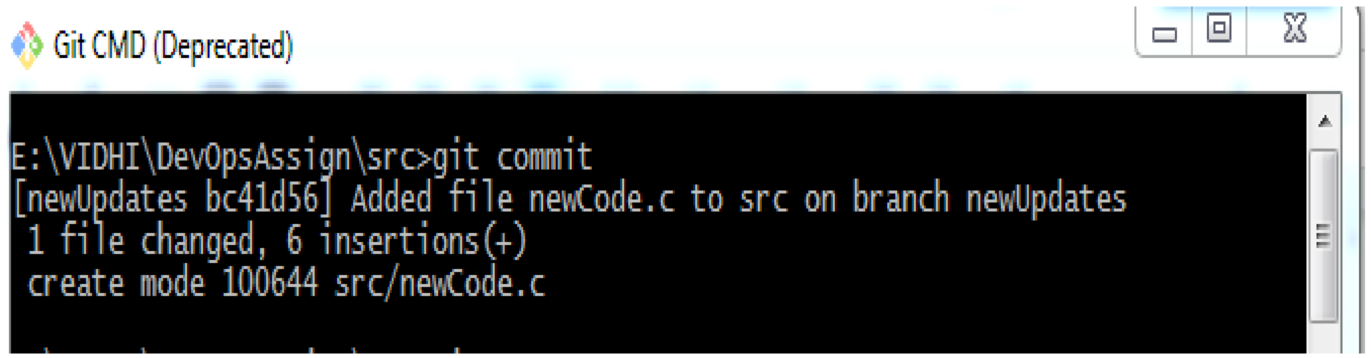
```
git add newCode.c
```



```
E:\VIDHI\DevOpsAssign\src>git add newCode.c

E:\VIDHI\DevOpsAssign\src>git status
On branch newUpdates
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   newCode.c
```

Commit using :

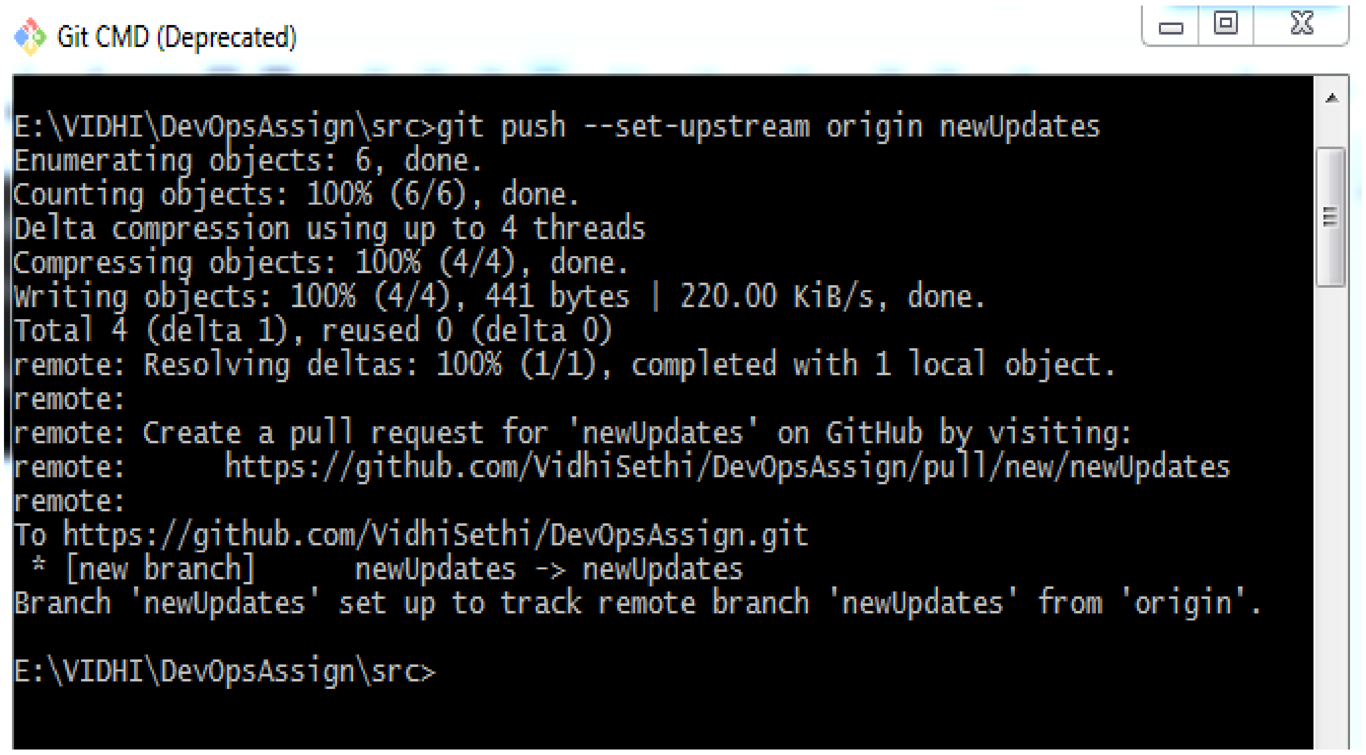
```
git commit
```



```
E:\VIDHI\DevOpsAssign\src>git commit
[newUpdates bc41d56] Added file newCode.c to src on branch newUpdates
 1 file changed, 6 insertions(+)
 create mode 100644 src/newCode.c
```

Copy changes to the remote repository using :

```
git push -set -upstream origin newUpdates
```



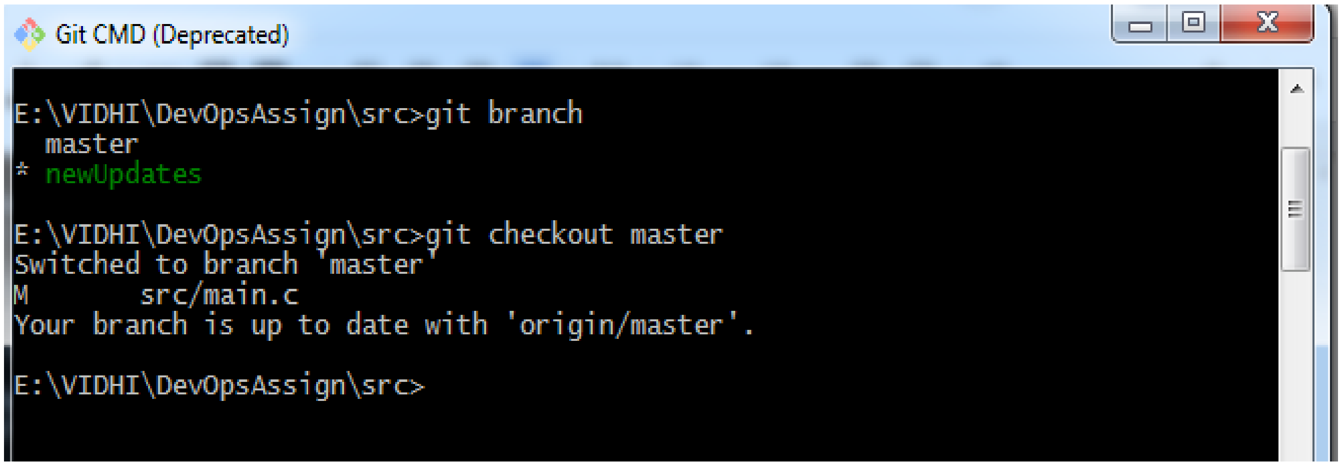
Git CMD (Deprecated)

```
E:\VIDHI\DevOpsAssign\src>git push --set-upstream origin newUpdates
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 441 bytes | 220.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'newUpdates' on GitHub by visiting:
remote:      https://github.com/VidhiSethi/DevOpsAssign/pull/new/newUpdates
remote:
To https://github.com/VidhiSethi/DevOpsAssign.git
 * [new branch]      newUpdates -> newUpdates
Branch 'newUpdates' set up to track remote branch 'newUpdates' from 'origin'.

E:\VIDHI\DevOpsAssign\src>
```

vii) Merge the Branch with Main line

First you need to checkout the branch you wish to merge into, for example here we will be merging in Master Branch.

```
git checkout master
```

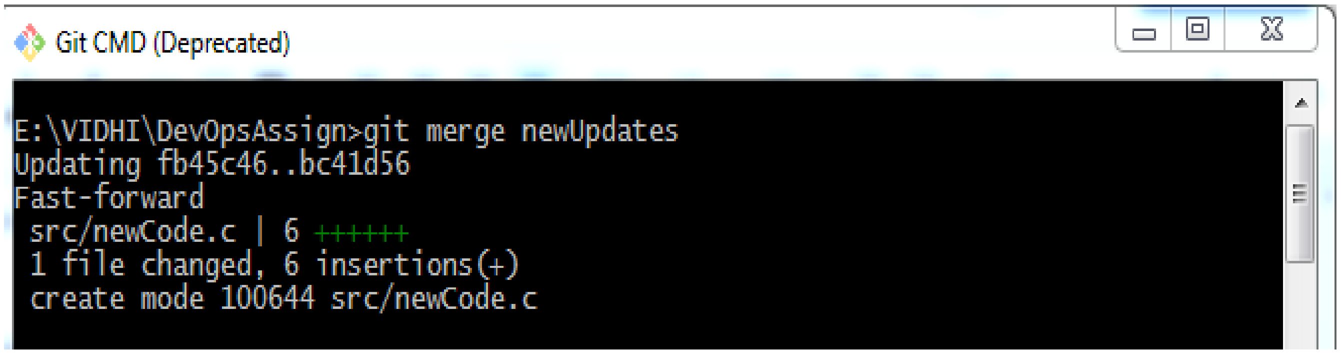


Git CMD (Deprecated)

```
E:\VIDHI\DevOpsAssign\src>git branch
  master
* newUpdates

E:\VIDHI\DevOpsAssign\src>git checkout master
Switched to branch 'master'
M       src/main.c
Your branch is up to date with 'origin/master'.

E:\VIDHI\DevOpsAssign\src>
```

Then Run :

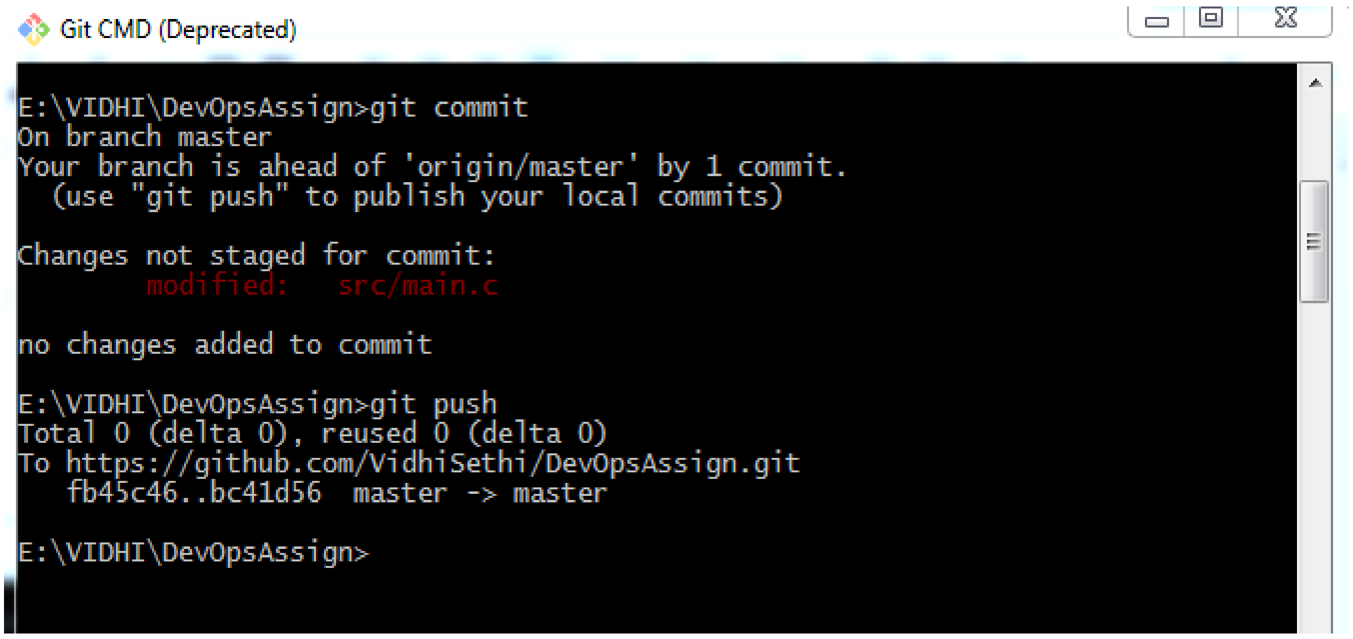
```
#git merge <name of branch>

git merge newUpdates
```



Git CMD (Deprecated)

```
E:\VIDHI\DevOpsAssign>git merge newUpdates
Updating fb45c46..bc41d56
Fast-forward
 src/newCode.c | 6 ++++++
 1 file changed, 6 insertions(+)
 create mode 100644 src/newCode.c
```

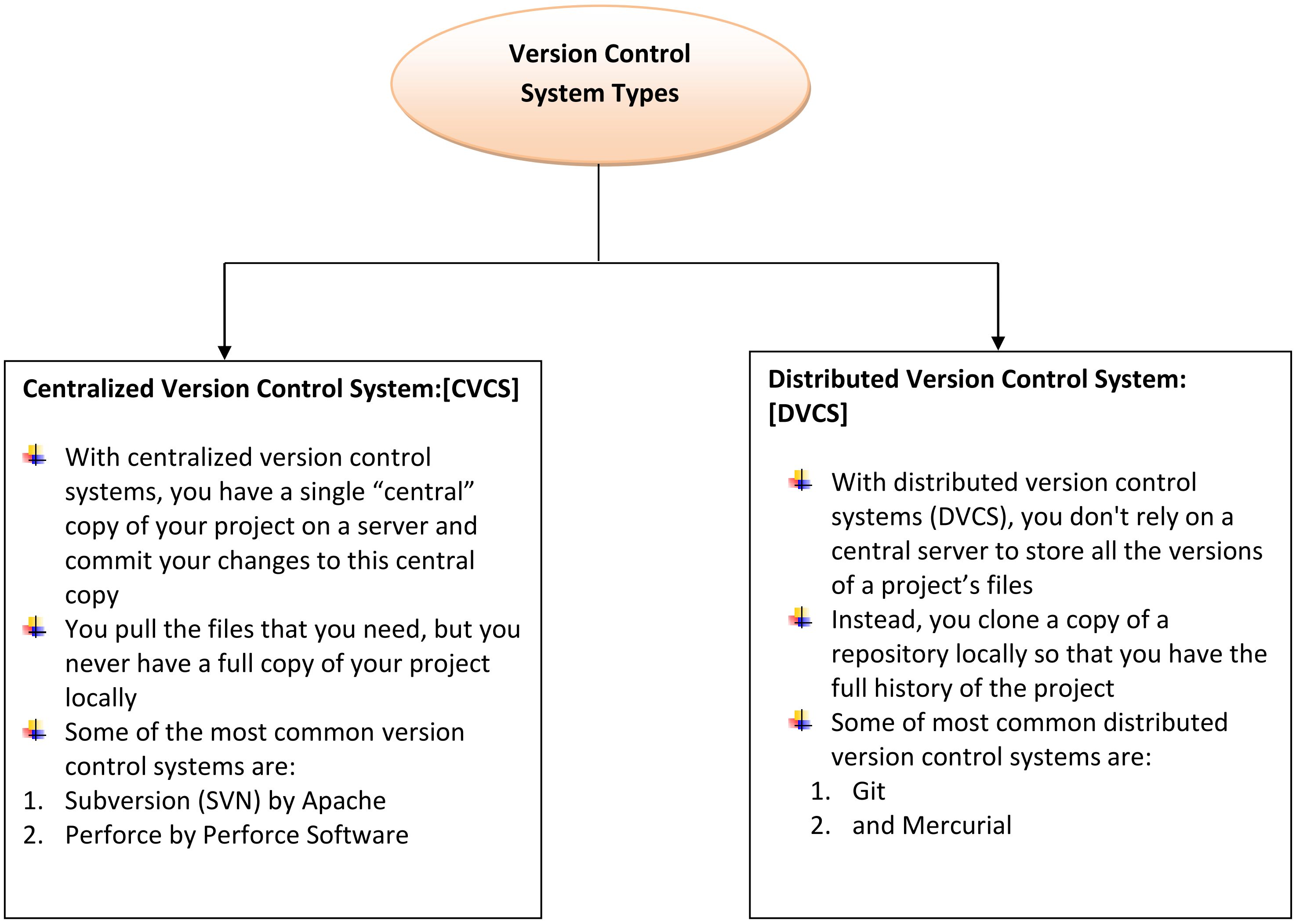
Commit and then push the changes to the remote repository.



Git CMD (Deprecated)

```
E:\VIDHI\DevOpsAssign>git commit
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
        modified:   src/main.c

no changes added to commit

E:\VIDHI\DevOpsAssign>git push
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/VidhiSethi/DevOpsAssign.git
   fb45c46..bc41d56  master -> master

E:\VIDHI\DevOpsAssign>
```

viii) Summary of advantages of moving from Centralized Source Code to Distributed Version Control



Centralized Version Control System:[CVCS]

- With centralized version control systems, you have a single “central” copy of your project on a server and commit your changes to this central copy
- You pull the files that you need, but you never have a full copy of your project locally
- Some of the most common version control systems are:

1. Subversion (SVN) by Apache
2. Perforce by Perforce Software

Distributed Version Control System: [DVCS]

- With distributed version control systems (DVCS), you don't rely on a central server to store all the versions of a project’s files
- Instead, you clone a copy of a repository locally so that you have the full history of the project
- Some of most common distributed version control systems are:

1. Git
2. and Mercurial

Advantages of moving from Centralized Source Code to Distributed Version Control are as follows:

1. Performing actions other than pushing and pulling change sets is extremely fast because the tool only needs to access the hard drive, not a remote server.
2. Everything but pushing and pulling can be done offline
3. Since each programmer has a full copy of the project repository, they can share changes with one or two other people at a time if they want to get some feedback
4. Allows users to work productively when not connected to a network.
5. Allows private work, so users can use their changes even for early drafts they do not want to publish.
6. Working copies effectively function as remote backups, which avoids relying on one physical machine as a single point of failure.
7. Allows various development models to be used, such as using development branches or a Commander/Lieutenant model.
8. On FOSS(Free and open-source software) software projects it is much easier to create a project fork from a project that is stalled because of leadership conflicts or design disagreements.
9. In Centralized Version Control Systems(CVCS) ,if the main server goes down, developers can’t save versioned changes.
10. In CVCS, if the central database is corrupted, the entire history could be lost (security issues).

*****