

ASSIGNMENT-2(INTRODUCTION TO GIT)

1. Git setup

```
ttn@ttn:~$ sudo apt-get install git
[sudo] password for ttn:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 429 not upgraded.
Need to get 4,733 kB of archives.
After this operation, 33.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 liberror-perl all 0.17025-1 [22.8 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git-man all 1:2.17.1-1ubuntu0.4 [803 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1:2.17.1-1ubuntu0.4 [3,907 kB]
Fetched 4,733 kB in 3s (1,812 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 128539 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17025-1_all.deb ...
Unpacking liberror-perl (0.17025-1) ...
```

2. Initialize a Git Repository

```
ttn@ttn:~/learnings$ git rebase --help
ttn@ttn:~/learnings$ git init
Reinitialized existing Git repository in /home/ttn/learnings/.git/
ttn@ttn:~/learnings$
```

3. Add files to the repository

```
ttn@ttn:~/learnings$ git add .
```

4. Unstage 1 file.

```
ttn@ttn:~/learnings$ git add .
ttn@ttn:~/learnings$ git status
On branch deepanshu
nothing to commit, working tree clean
ttn@ttn:~/learnings$ git diff
ttn@ttn:~/learnings$ git status
On branch deepanshu
nothing to commit, working tree clean
ttn@ttn:~/learnings$
```

5. Commit the file

```
ttn@ttn:~/learnings$ git commit -m "first file committed"
[master (root-commit) e50bc03] first file committed
1 file changed, 1 insertion(+)
create mode 100644 learninggit.txt
```

6. Add a remote

```
ttn@ttn:~$ git clone git@github.com:shobhitttn/learnings.git
Cloning into 'learnings'...
warning: You appear to have cloned an empty repository.
```

7. Undo changes to a particular file

```
ttn@ttn:~/learnings$ git checkout ea567d7ca6932082e958c51df0b7cd02b422c3b0
Note: checking out 'ea567d7ca6932082e958c51df0b7cd02b422c3b0'.

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 ea567d7 fdgfdgfdgfdffdgdg
ttn@ttn:~/learnings$
```

8. Push changes to Github

```
ttn@ttn:~/learnings$ git push origin deepanshu
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 292 bytes | 292.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:shobhitttn/learnings.git
   1fffaab..0321f85  deepanshu -> deepanshu
```

9. Clone the repository

```
ttn@ttn:~$ git clone git@github.com:shobhitttn/learnings.git
Cloning into 'learnings'...
warning: You appear to have cloned an empty repository.
```

10. Add changes to one of the copies and pull the changes in the other.

```

ttn@ttn:~/learnings$ ls
harshdeep.txt  hello.txt  tt.txt
ttn@ttn:~/learnings$ cat tt.txt
ihsgdhgsjdgjgsdj
ttn@ttn:~/learnings$ vi tt.txt
ttn@ttn:~/learnings$ cat tt.txt

Cdmdnmddmbdmd
cbcbccb
ccbccbcm
chkdkdd
A
ihsgdhgsjdgjgsdj
ttn@ttn:~/learnings$ git add .
ttn@ttn:~/learnings$ git commit -m "another committed"
[detached HEAD 0e38a24] another committed
1 file changed, 6 insertions(+)
ttn@ttn:~/learnings$ git push origin deepanshu
Warning: Permanently added the RSA host key for IP address '140.82.118.3' to the list of known hosts.
Everything up-to-date
ttn@ttn:~/learnings$ █

```

11. Check differences between a file and its staged version

```

ttn@ttn:~/learnings$ vi tt.txt
ttn@ttn:~/learnings$ git add tt.txt
ttn@ttn:~/learnings$ git diff --staged
diff --git a/tt.txt b/tt.txt
index b9ce139..8bd3e6c 100644
--- a/tt.txt
+++ b/tt.txt
@@ -1,4 +1,4 @@
-
+ddgkjdgdkgdjkgdkgdjkgdkgdg
Cdmdnmddmbdmd
cbcbccb
ccbccbcm
ttn@ttn:~/learnings$ █

```

12. Ignore a few files to be checked in

```

ttn@ttn:~/learnings$ touch .gitignore
ttn@ttn:~/learnings$ vi .gitignore
ttn@ttn:~/learnings$ cat .gitignore
ttn@ttn:~/learnings$ vi .gitignore
ttn@ttn:~/learnings$ vi nottobepushed.txt
ttn@ttn:~/learnings$ git add .gitignore
ttn@ttn:~/learnings$ git status
HEAD detached from ea567d7
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   .gitignore
        modified:   tt.txt

ttn@ttn:~/learnings$ git add nottobepushed.txt
The following paths are ignored by one of your .gitignore files:
nottobepushed.txt
Use -f if you really want to add them.
ttn@ttn:~/learnings$ █

```

13. Create a new branch

```

ttn@ttn:~/learnings$ git checkout -b newbranch
Switched to a new branch 'newbranch'
ttn@ttn:~/learnings$ git branch
  abc
  deepanshu
  hello
  master
* newbranch
ttn@ttn:~/learnings$ █

```

14. Diverge them with commits


```
Terminal
File Edit View Search Terminal Tabs Help

ttn@ttn:~/learnings$ ls
file1 harshdeep.txt hello.txt learninggit.txt nottobepushed.txt tt.txt xyz.txt
ttn@ttn:~/learnings$ git checkout newbranch
Switched to branch 'newbranch'
ttn@ttn:~/learnings$ ls
file1 harshdeep.txt hello.txt tt.txt xyz.txt
ttn@ttn:~/learnings$ rm xyz.txt
ttn@ttn:~/learnings$ ls
file1 harshdeep.txt hello.txt tt.txt
ttn@ttn:~/learnings$ git add .
ttn@ttn:~/learnings$ git commit -m "deleted xyz.txt"
[newbranch 6bce26c] deleted xyz.txt
 1 file changed, 3 deletions(-)
 delete mode 100644 xyz.txt
ttn@ttn:~/learnings$ git push origin newbranch
Counting objects: 2, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 221 bytes | 221.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:shobhitn/learnings.git
   bf65b08..6bce26c newbranch -> newbranch
ttn@ttn:~/learnings$ git checkout deepanshu
Switched to branch 'deepanshu'
ttn@ttn:~/learnings$ git pull origin newbranch
From github.com:shobhitn/learnings
 * branch                newbranch -> FETCH_HEAD
Removing xyz.txt
Merge made by the 'recursive' strategy.
 xyz.txt | 3 ---
 1 file changed, 3 deletions(-)
 delete mode 100644 xyz.txt
ttn@ttn:~/learnings$ ls
file1 harshdeep.txt hello.txt learninggit.txt nottobepushed.txt tt.txt
ttn@ttn:~/learnings$
```

15. Edit the same file at the same line on both branches and commit

```
Terminal
File Edit View Search Terminal Tabs Help

ttn@ttn:~/learnings$ ls
harshdeep.txt hello.txt tt.txt
ttn@ttn:~/learnings$ vi harshdeep.txt
ttn@ttn:~/learnings$ git add .
ttn@ttn:~/learnings$ git commit -m "harsh changed"
[newbranch 5ed3011] harsh changed
 1 file changed, 1 insertion(+)
ttn@ttn:~/learnings$ git push origin newbranch
To github.com:shobhitn/learnings.git
 ! [rejected]        newbranch -> newbranch (fetch first)
error: failed to push some refs to 'git@github.com:shobhitn/learnings.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
ttn@ttn:~/learnings$ git pull origin newbranch
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Total 6 (delta 4), reused 4 (delta 4), pack-reused 2
Unpacking objects: 100% (6/6), done.
From github.com:shobhitn/learnings
 * branch                newbranch -> FETCH_HEAD
 * [new branch]          newbranch -> origin/newbranch
Merge made by the 'recursive' strategy.
 file1 | 3 +++
 hello.txt | 23 ++++++++++++++++++++++
 xyz.txt | 3 +++
 3 files changed, 29 insertions(+)
 create mode 100644 file1
 create mode 100644 xyz.txt
ttn@ttn:~/learnings$ git push origin newbranch
Counting objects: 12, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (11/11), done.
Writing objects: 100% (12/12), 1.34 KiB | 1.34 MiB/s, done.
Total 12 (delta 4), reused 0 (delta 0)
```

16. Try merging and resolve merge conflicts

```
Terminal
File Edit View Search Terminal Tabs Help
Terminal
Auto-merging harshdeep.txt
CONFLICT (content): Merge conflict in harshdeep.txt
Automatic merge failed; fix conflicts and then commit the result.
ttn@ttn:~/learnings$ vi harshdeep.txt
ttn@ttn:~/learnings$ git add harshdeep.txt
ttn@ttn:~/learnings$ git commit -m "conflict resolved"
U
hello.txt
error: Committing is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
ttn@ttn:~/learnings$ vi harshdeep.txt
ttn@ttn:~/learnings$ git add harshdeep.txt
ttn@ttn:~/learnings$ git commit -m "conflict resolved"
U
hello.txt
error: Committing is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
ttn@ttn:~/learnings$ vim hello.txt

Command 'vim' not found, but can be installed with:

sudo apt install vim
sudo apt install vim-gtk3
sudo apt install vim-tiny
sudo apt install neovim
sudo apt install vim-athena
sudo apt install vim-gtk
sudo apt install vim-nox

ttn@ttn:~/learnings$ vi hello.txt
ttn@ttn:~/learnings$ git add .
ttn@ttn:~/learnings$ git commit -m "committing"
[deepanshu ea7c4b8] committing
ttn@ttn:~/learnings$
```

17. Stash the changes and pop them.

```
Terminal
File Edit View Search Terminal Tabs Help
Terminal
ttn@ttn:~/learnings$ vi hello.txt
ttn@ttn:~/learnings$ git stash
Saved working directory and index state WIP on deepanshu: 0321f85 heeleleo
ttn@ttn:~/learnings$ git commit -m "next commit"
On branch deepanshu
Untracked files:
  nottobepushed.txt

nothing added to commit but untracked files present
ttn@ttn:~/learnings$ git status
On branch deepanshu
Untracked files:
  (use "git add <file>..." to include in what will be committed)

  nottobepushed.txt

nothing added to commit but untracked files present (use "git add" to track)
ttn@ttn:~/learnings$ git push origin deepanshu
Everything up-to-date
ttn@ttn:~/learnings$ git stash pop
On branch deepanshu
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   hello.txt
    modified:   learninggit.txt

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

  nottobepushed.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (afb56b6b577e4b19298389e6d1e88260574b31d9)
ttn@ttn:~/learnings$
```

18. Add the following code to your `.bashrc` file : `color_prompt="yes"`

```

parse_git_branch() {
git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*\)/(\1)/'
}
if [ "$color_prompt" = yes ]; then
PS1='\u@\h[\033[00m\]:[\033[01;34m]\W[\033[01;31m]
$(parse_git_branch)\[\033[00m\]\$ '
else
PS1='\u@\h:\W $(parse_git_branch)\$ '
fi
unset color_prompt force_color_prompt

```

```
ttn@ttn:learnings (deepanshu)$  
ttn@ttn:learnings (deepanshu)$ git branch  
  abc  
* deepanshu  
  hello  
  master  
  newbranch  
ttn@ttn:learnings (deepanshu)$ git checkout hello  
Switched to branch 'hello'  
Your branch and 'origin/hello' have diverged,  
and have 4 and 7 different commits each, respectively.  
(use "git pull" to merge the remote branch into yours)  
ttn@ttn:learnings (hello)$
```