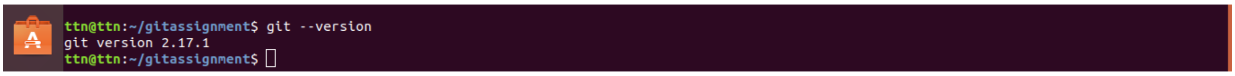
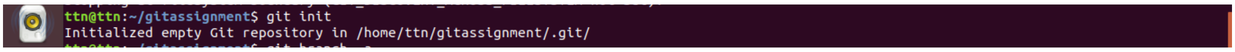
1. Git Setup



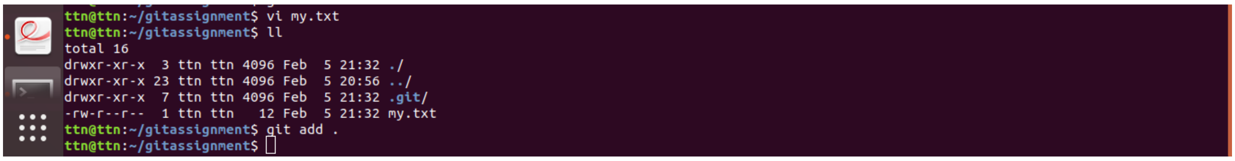
```
ttn@ttn:~/gitassignment$ git --version
git version 2.17.1
ttn@ttn:~/gitassignment$ 
```

2. Initialize a Git Repository



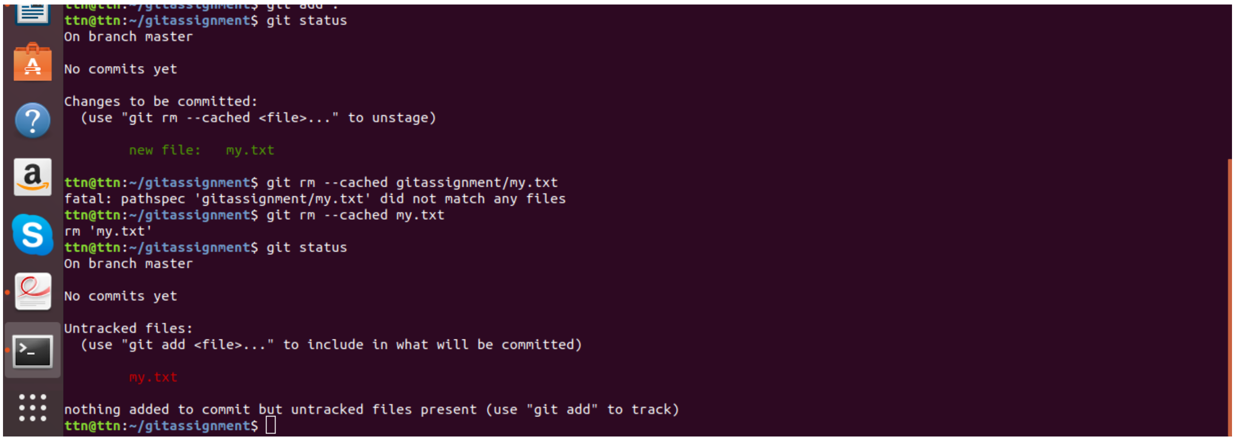
```
ttn@ttn:~/gitassignment$ git init
Initialized empty Git repository in /home/ttn/gitassignment/.git/
```

3. Add files to the repository



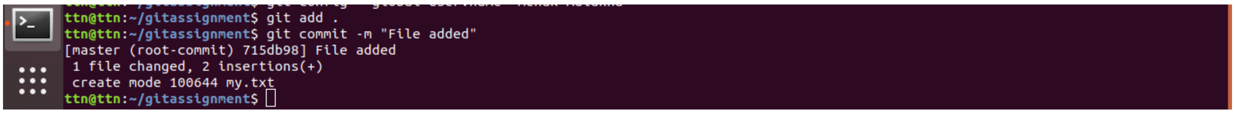
```
ttn@ttn:~/gitassignment$ vi my.txt
ttn@ttn:~/gitassignment$ ll
total 16
drwxr-xr-x  3 ttn ttn 4096 Feb  5 21:32 ./
drwxr-xr-x 23 ttn ttn 4096 Feb  5 20:56 ../
drwxr-xr-x  7 ttn ttn 4096 Feb  5 21:32 .git/
-rw-r--r--  1 ttn ttn   12 Feb  5 21:32 my.txt
ttn@ttn:~/gitassignment$ git add .
ttn@ttn:~/gitassignment$ 
```

4. Unstage 1 file



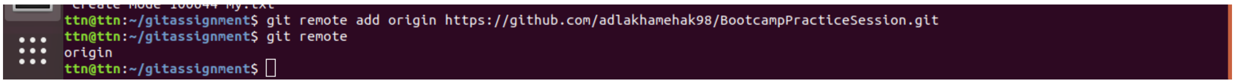
```
ttn@ttn:~/gitassignment$ git add .
ttn@ttn:~/gitassignment$ git status
On branch master

No commits yet

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

        new file:   my.txt

ttn@ttn:~/gitassignment$ git rm --cached gitassignment/my.txt
fatal: pathspec 'gitassignment/my.txt' did not match any files
ttn@ttn:~/gitassignment$ git rm --cached my.txt
rm 'my.txt'
ttn@ttn:~/gitassignment$ git status
On branch master

No commits yet

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

        my.txt

nothing added to commit but untracked files present (use "git add" to track)
ttn@ttn:~/gitassignment$ 
```

5. Commit the file



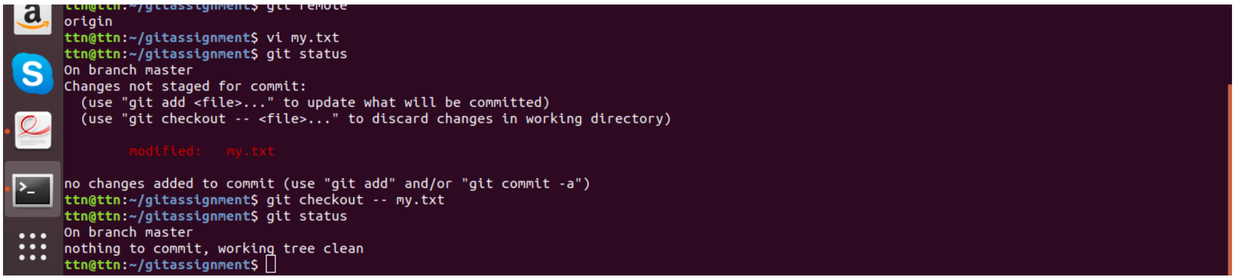
```
ttn@ttn:~/gitassignment$ git add .
ttn@ttn:~/gitassignment$ git commit -m "File added"
[master (root-commit) 715db98] File added
 1 file changed, 2 insertions(+)
 create mode 100644 my.txt
ttn@ttn:~/gitassignment$ 
```

6. Add a remote



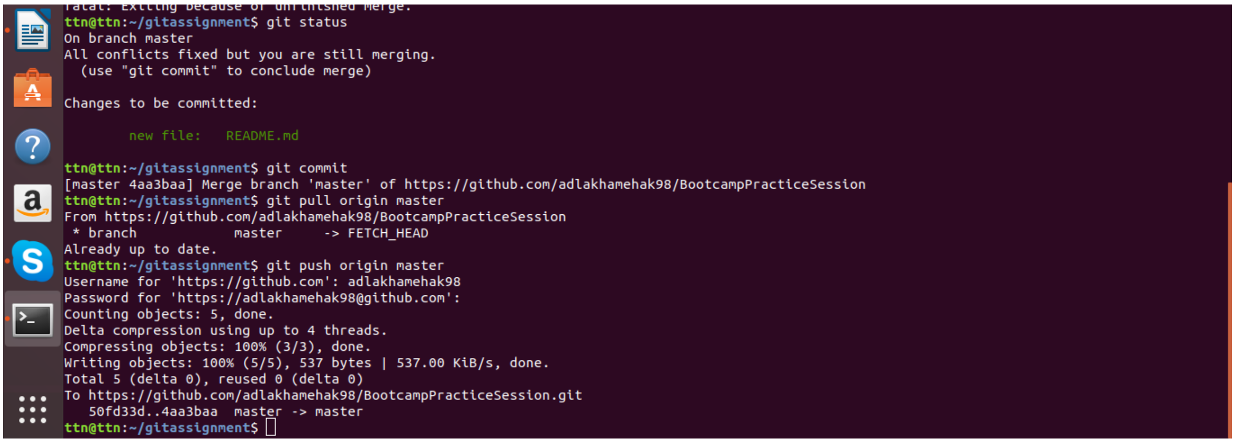
```
 create mode 100644 my.txt
ttn@ttn:~/gitassignment$ git remote add origin https://github.com/adlakhamehak98/BootcampPracticeSession.git
ttn@ttn:~/gitassignment$ git remote
origin
ttn@ttn:~/gitassignment$ 
```

7. Undo changes to a particular file



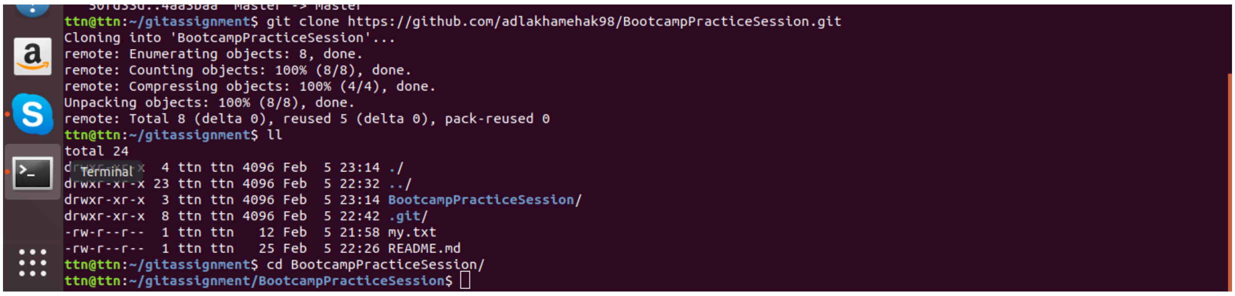
```
ttn@ttn:~/gitassignment$ git remote
origin
ttn@ttn:~/gitassignment$ vi my.txt
ttn@ttn:~/gitassignment$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   my.txt

no changes added to commit (use "git add" and/or "git commit -a")
ttn@ttn:~/gitassignment$ git checkout -- my.txt
ttn@ttn:~/gitassignment$ git status
On branch master
nothing to commit, working tree clean
ttn@ttn:~/gitassignment$ 
```

8. Push changes to Github



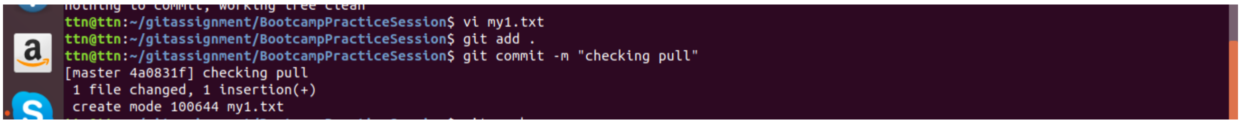
```
fatal: Exiting because of unfinished merge.
ttn@ttn:~/gitassignment$ git status
On branch master
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:

        new file:   README.md

ttn@ttn:~/gitassignment$ git commit
[master 4aa3baa] Merge branch 'master' of https://github.com/adlakhamehak98/BootcampPracticeSession
ttn@ttn:~/gitassignment$ git pull origin master
From https://github.com/adlakhamehak98/BootcampPracticeSession
 * branch            master     -> FETCH_HEAD
Already up to date.
ttn@ttn:~/gitassignment$ git push origin master
Username for 'https://github.com': adlakhamehak98
Password for 'https://adlakhamehak98@github.com':
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 537 bytes | 537.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To https://github.com/adlakhamehak98/BootcampPracticeSession.git
   50fd33d..4aa3baa  master -> master
ttn@ttn:~/gitassignment$ 
```

9. Clone the repository

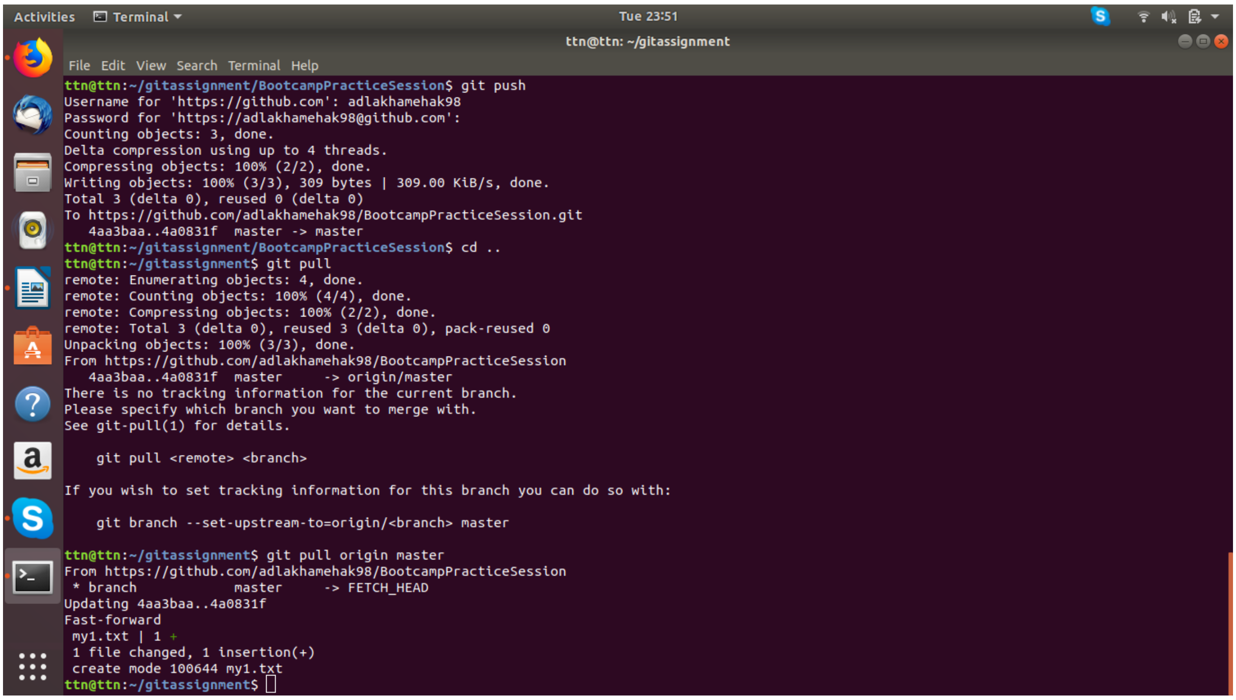


```
   50fd33d..4aa3baa  master -> master
ttn@ttn:~/gitassignment$ git clone https://github.com/adlakhamehak98/BootcampPracticeSession.git
Cloning into 'BootcampPracticeSession'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (4/4), done.
Unpacking objects: 100% (8/8), done.
remote: Total 8 (delta 0), reused 5 (delta 0), pack-reused 0
ttn@ttn:~/gitassignment$ ll
total 24
drwxr-xr-x  4 ttn ttn 4096 Feb  5 23:14 ./
drwxr-xr-x 23 ttn ttn 4096 Feb  5 22:32 ../
drwxr-xr-x  3 ttn ttn 4096 Feb  5 23:14 BootcampPracticeSession/
drwxr-xr-x  8 ttn ttn 4096 Feb  5 22:42 .git/
-rw-r--r--  1 ttn ttn   12 Feb  5 21:58 my.txt
-rw-r--r--  1 ttn ttn   25 Feb  5 22:26 README.md
ttn@ttn:~/gitassignment$ cd BootcampPracticeSession/
ttn@ttn:~/gitassignment/BootcampPracticeSession$ 
```

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

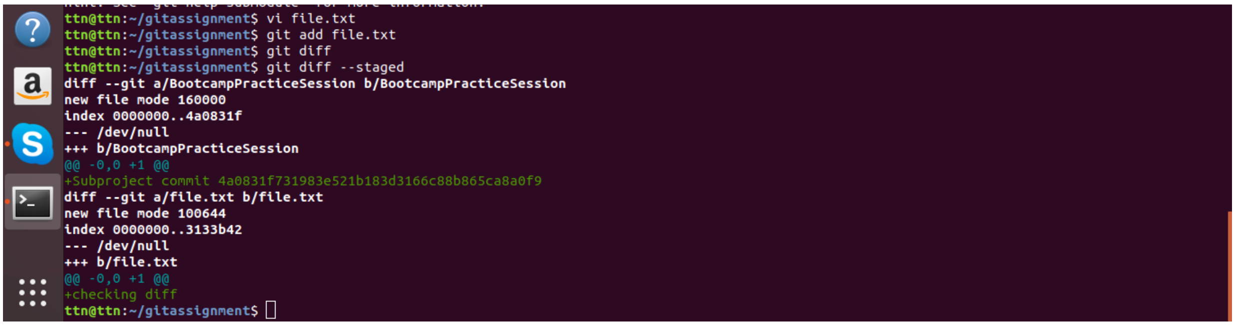


```
nothing to commit, working tree clean
ttn@ttn:~/gitassignment/BootcampPracticeSession$ vi my1.txt
ttn@ttn:~/gitassignment/BootcampPracticeSession$ git add .
ttn@ttn:~/gitassignment/BootcampPracticeSession$ git commit -m "checking pull"
[master 4a0831f] checking pull
 1 file changed, 1 insertion(+)
 create mode 100644 my1.txt
```



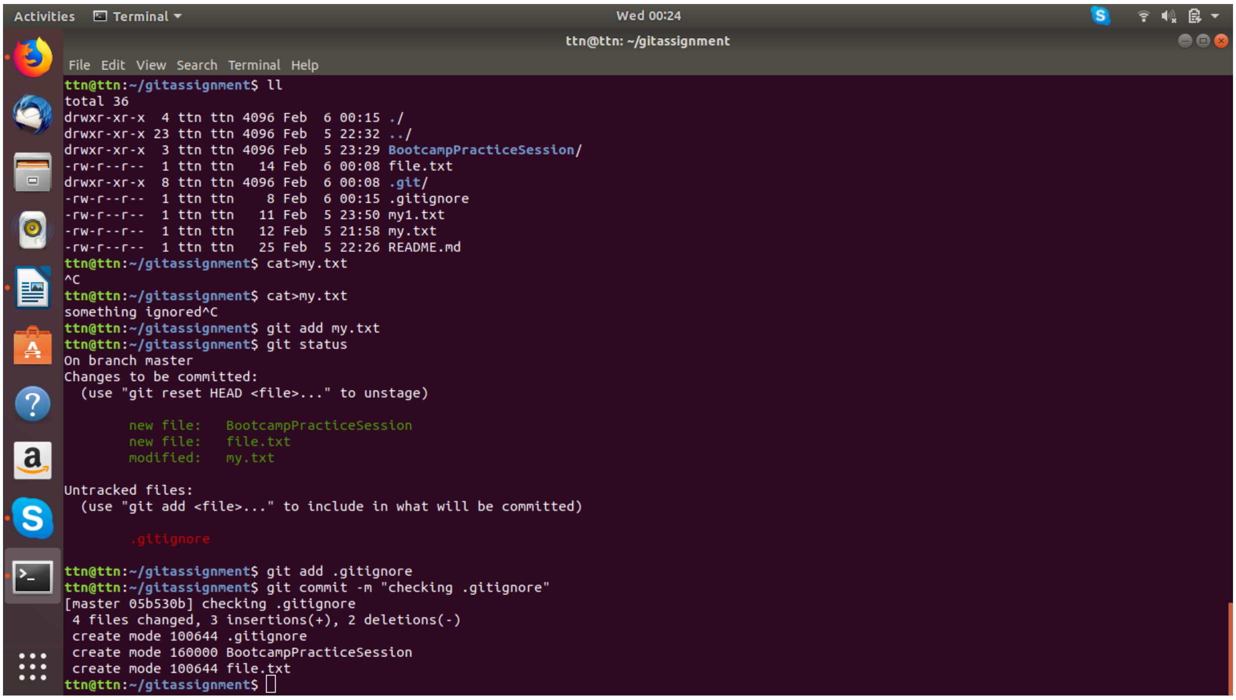
```
ttn@ttn:~/gitassignment/BootcampPracticeSession$ git push
Username for 'https://github.com': adlakhamehak98
Password for 'https://adlakhamehak98@github.com':
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 309 bytes | 309.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/adlakhamehak98/BootcampPracticeSession.git
   4aa3baa..4a0831f  master -> master
ttn@ttn:~/gitassignment/BootcampPracticeSession$ cd ..
ttn@ttn:~/gitassignment$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/adlakhamehak98/BootcampPracticeSession
   4aa3baa..4a0831f  master     -> origin/master
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=origin/<branch> master

ttn@ttn:~/gitassignment$ git pull origin master
From https://github.com/adlakhamehak98/BootcampPracticeSession
 * branch            master     -> FETCH_HEAD
Updating 4aa3baa..4a0831f
Fast-forward
 my1.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 my1.txt
ttn@ttn:~/gitassignment$ 
```

11.Check differences between a file and its staged version



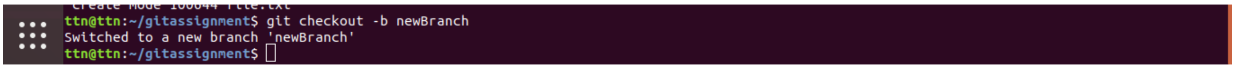
```
ttn@ttn:~/gitassignment$ vi file.txt
ttn@ttn:~/gitassignment$ git add file.txt
ttn@ttn:~/gitassignment$ git diff
ttn@ttn:~/gitassignment$ git diff --staged
diff --git a/BootcampPracticeSession b/BootcampPracticeSession
new file mode 160000
index 0000000..4a0831f
--- /dev/null
+++ b/BootcampPracticeSession
@@ -0,0 +1 @@
+Subproject commit 4a0831f731983e521b183d3166c88b865ca8a0f9
diff --git a/file.txt b/file.txt
new file mode 100644
index 0000000..3133b42
--- /dev/null
+++ b/file.txt
@@ -0,0 +1 @@
+checking diff
ttn@ttn:~/gitassignment$
```

12.Ignore a few files to be checked in



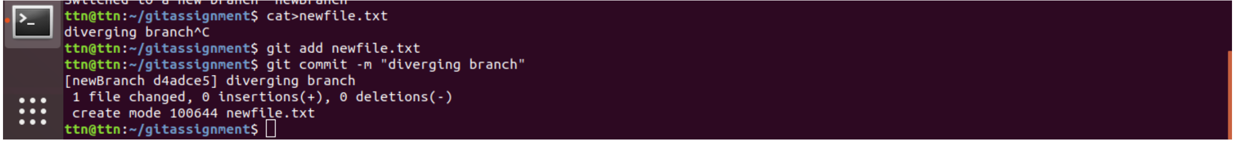
```
ttn@ttn:~/gitassignment$ ll
total 36
drwxr-xr-x  4 ttn ttn 4096 Feb  6 00:15 ./
drwxr-xr-x 23 ttn ttn 4096 Feb  5 22:32 ../
drwxr-xr-x  3 ttn ttn 4096 Feb  5 23:29 BootcampPracticeSession/
-rw-r--r--  1 ttn ttn   14 Feb  6 00:08 file.txt
drwxr-xr-x  8 ttn ttn 4096 Feb  6 00:08 .git/
-rw-r--r--  1 ttn ttn    8 Feb  6 00:15 .gitignore
-rw-r--r--  1 ttn ttn   11 Feb  5 23:50 my1.txt
-rw-r--r--  1 ttn ttn   12 Feb  5 21:58 my.txt
-rw-r--r--  1 ttn ttn   25 Feb  5 22:26 README.md
ttn@ttn:~/gitassignment$ cat>my.txt
^C
ttn@ttn:~/gitassignment$ cat>my.txt
something ignored^C
ttn@ttn:~/gitassignment$ git add my.txt
ttn@ttn:~/gitassignment$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   BootcampPracticeSession
        new file:   file.txt
        modified:   my.txt

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

        .gitignore

ttn@ttn:~/gitassignment$ git add .gitignore
ttn@ttn:~/gitassignment$ git commit -m "checking .gitignore"
[master 05b530b] checking .gitignore
 4 files changed, 3 insertions(+), 2 deletions(-)
 create mode 100644 .gitignore
 create mode 160000 BootcampPracticeSession
 create mode 100644 file.txt
ttn@ttn:~/gitassignment$
```

13.Create a new branch.



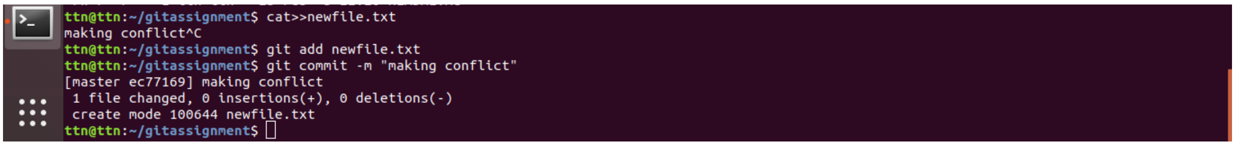
```
ttn@ttn:~/gitassignment$ git checkout -b newBranch
Switched to a new branch 'newBranch'
ttn@ttn:~/gitassignment$
```

14.Diverge them with commits



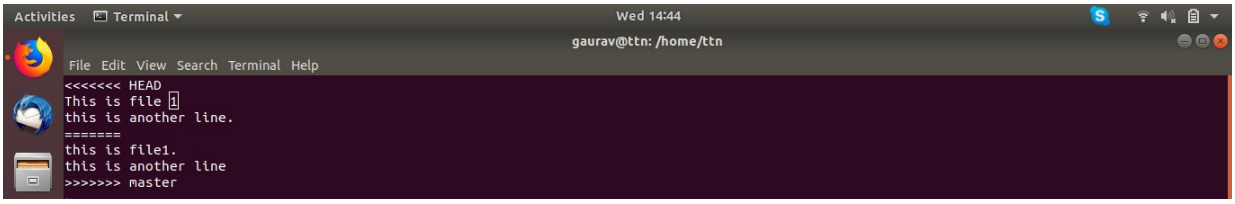
```
ttn@ttn:~/gitassignment$ cat>newfile.txt
diverging branch^C
ttn@ttn:~/gitassignment$ git add newfile.txt
ttn@ttn:~/gitassignment$ git commit -m "diverging branch"
[newBranch d4adce5] diverging branch
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 newfile.txt
ttn@ttn:~/gitassignment$
```

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



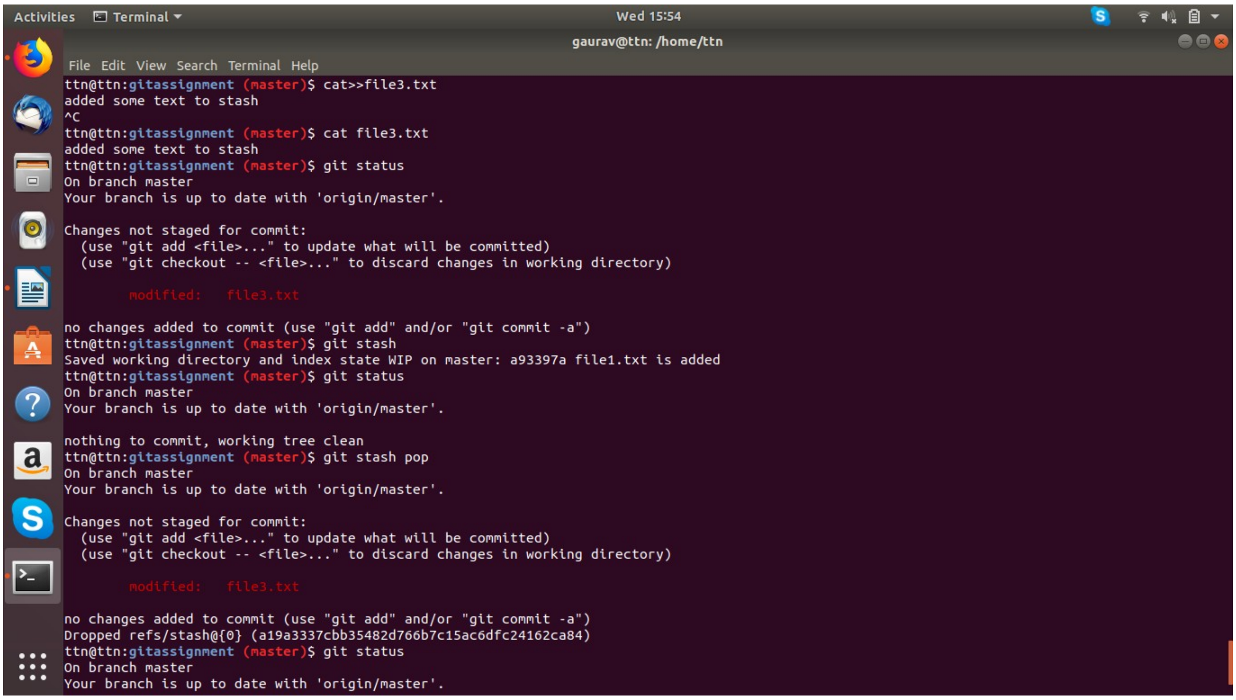
```
ttn@ttn:~/gitassignment$ cat>>newfile.txt
making conflict^C
ttn@ttn:~/gitassignment$ git add newfile.txt
ttn@ttn:~/gitassignment$ git commit -m "making conflict"
[master ec77169] making conflict
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 newfile.txt
ttn@ttn:~/gitassignment$
```

16.Try merging and resolve merge conflicts

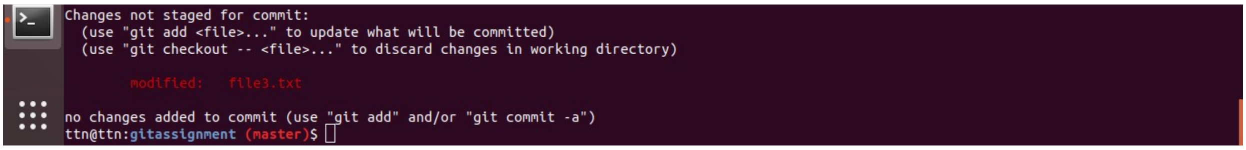


```
<<<<<<< HEAD
This is file 1
this is another line.
=======
this is file1.
this is another line
>>>>>>> master
```

17.Stash the changes and pop them



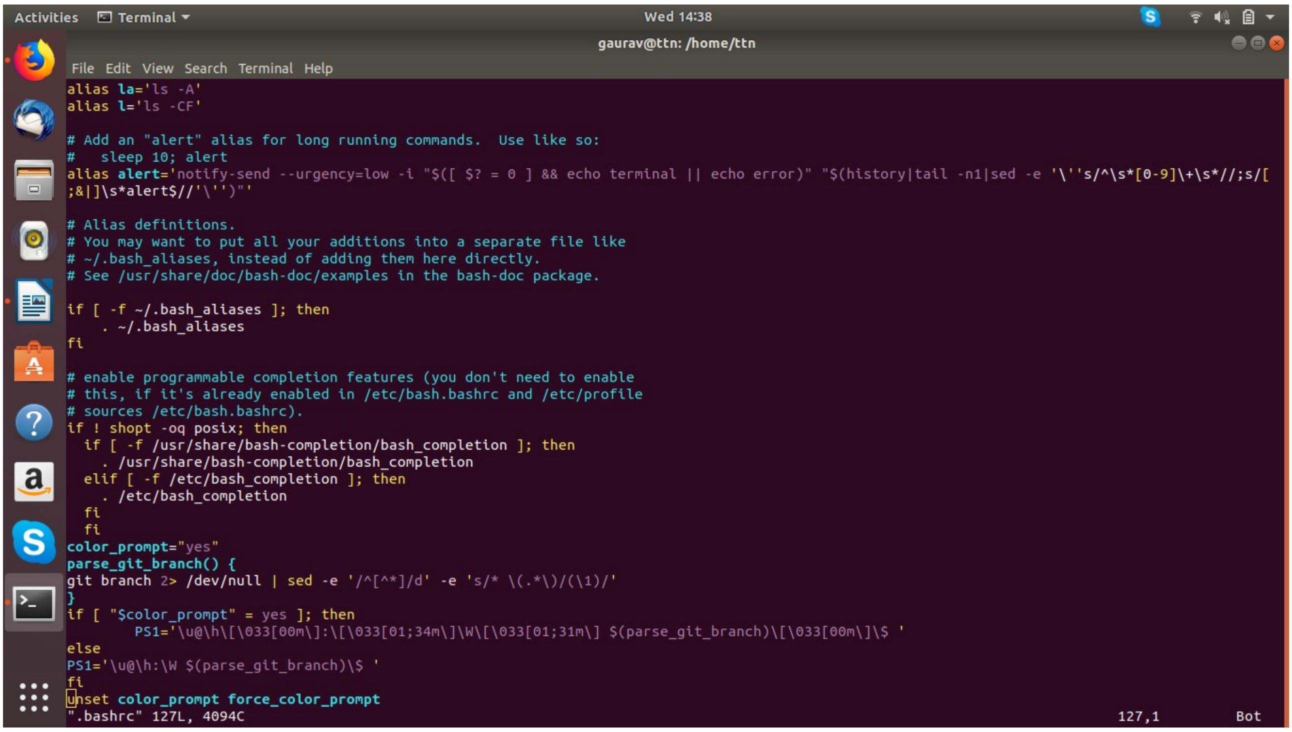
```
ttn@ttn:gitassignment (master)$ cat>>file3.txt
added some text to stash
^C
ttn@ttn:gitassignment (master)$ cat file3.txt
added some text to stash
ttn@ttn:gitassignment (master)$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   file3.txt

no changes added to commit (use "git add" and/or "git commit -a")
ttn@ttn:gitassignment (master)$ git stash
Saved working directory and index state WIP on master: a93397a file1.txt is added
ttn@ttn:gitassignment (master)$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
ttn@ttn:gitassignment (master)$ git stash pop
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 checkout -- <file>..." to discard changes in working directory)

        modified:   file3.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (a19a3337cbb35482d766b7c15ac6dfc24162ca84)
ttn@ttn:gitassignment (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 checkout -- <file>..." to discard changes in working directory)

        modified:   file3.txt

no changes added to commit (use "git add" and/or "git commit -a")
ttn@ttn:gitassignment (master)$ 
```

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[
else
PS1='\u@\h:\W $(parse_git_branch)\$ '
fi
unset color_prompt force_color_prompt
```



```
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[
;&|]\s*alert$//'\'')"'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
  fi
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
".bashrc" 127L, 4094C                                                                                    127,1         Bot
```