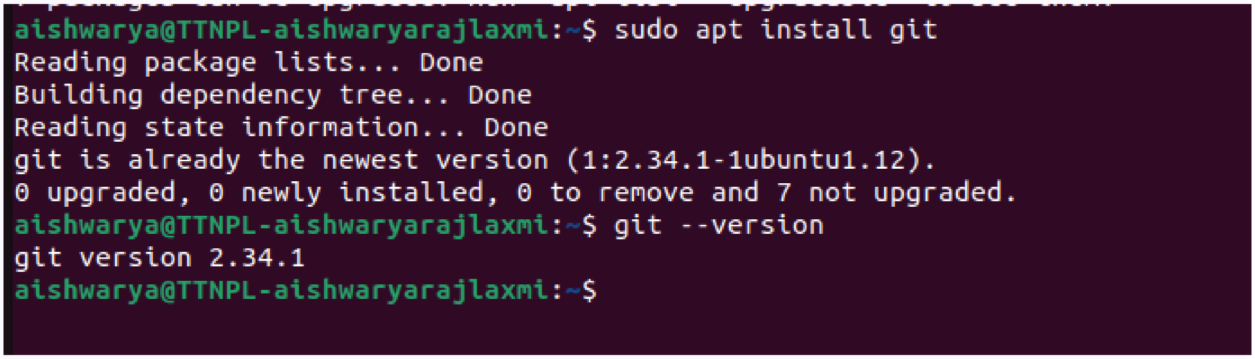
INTRODUCTION TO VERSION CONTROL

Q1)Git Setup
https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html

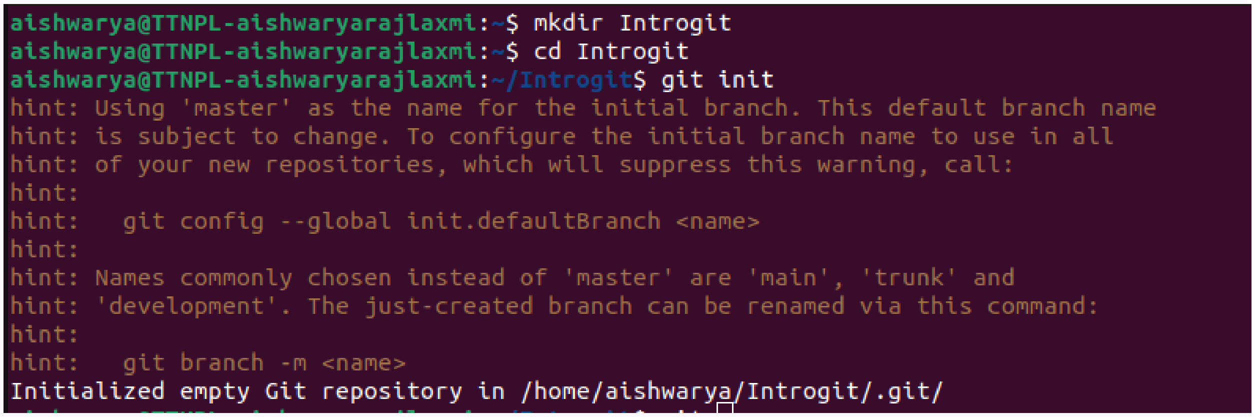
Sol:
Installing the git and making further changes



```
aishwarya@TTNPL-aishwaryarajlaxmi:~$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.12).
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
aishwarya@TTNPL-aishwaryarajlaxmi:~$ git --version
git version 2.34.1
aishwarya@TTNPL-aishwaryarajlaxmi:~$
```

Q2)Initialize a Git Repository

Sol:
firstly created a Introgit folder and went inside the folder, created a new Git repository in our project folder to start tracking changes.

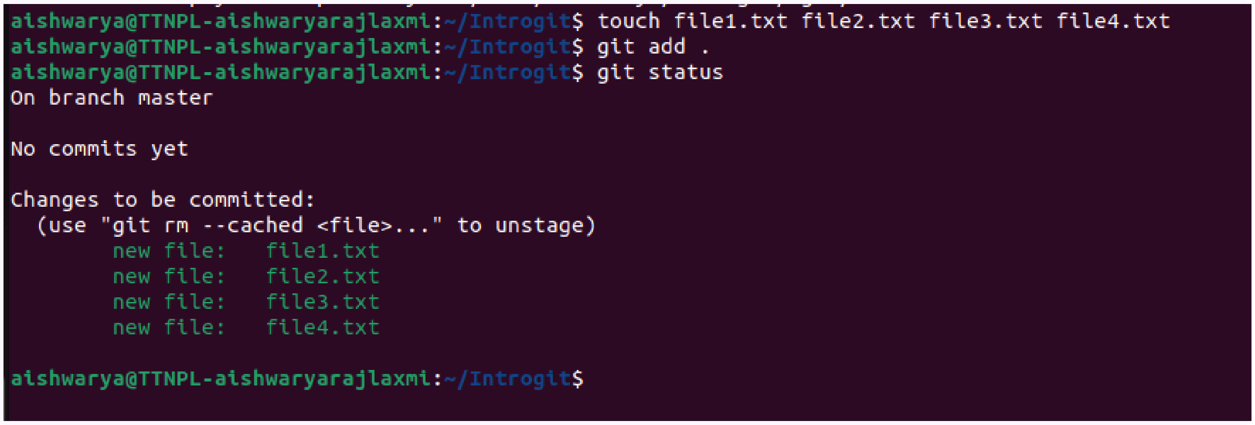


```
aishwarya@TTNPL-aishwaryarajlaxmi:~$ mkdir Introgit
aishwarya@TTNPL-aishwaryarajlaxmi:~$ cd Introgit
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/aishwarya/Introgit/.git/
```

Q3)Add files to the repository

Sol:

we can stage files for commit by using add command.

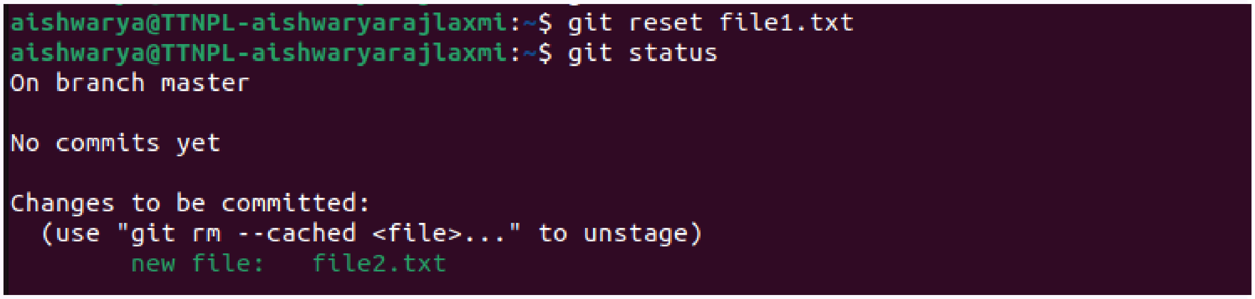


```
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ touch file1.txt file2.txt file3.txt file4.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git add .
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1.txt
        new file:   file2.txt
        new file:   file3.txt
        new file:   file4.txt

aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$
```

Q4)Unstage 1 file

Sol:

To unstage a file (but keep the changes) using the reset:

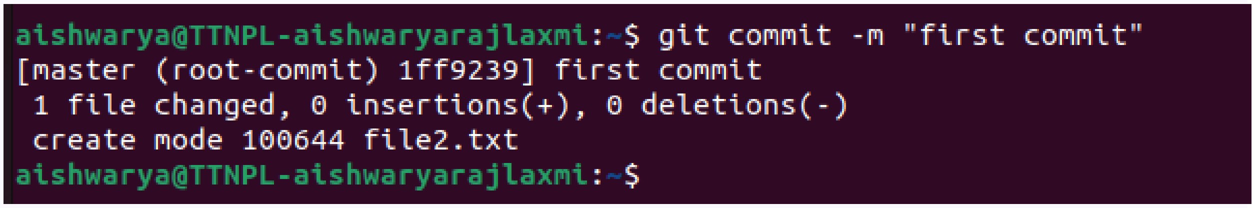


```
aishwarya@TTNPL-aishwaryarajlaxmi:~$ git reset file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file2.txt
```

Q5)Commit the file

Sol:

This saves the staged changes in the repository:

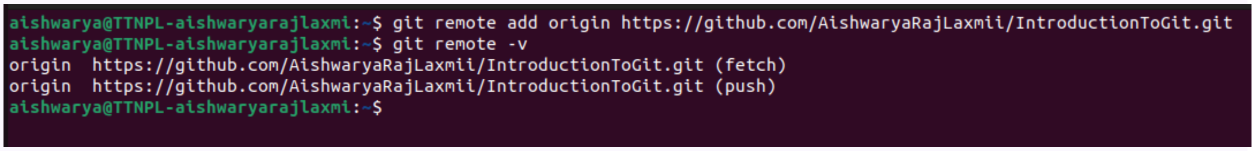


```
aishwarya@TTNPL-aishwaryarajlaxmi:~$ git commit -m "first commit"
[master (root-commit) 1ff9239] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file2.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~$
```

Q6)Add a remote

Sol:

Linking the local repository with a remote server (GitHub):

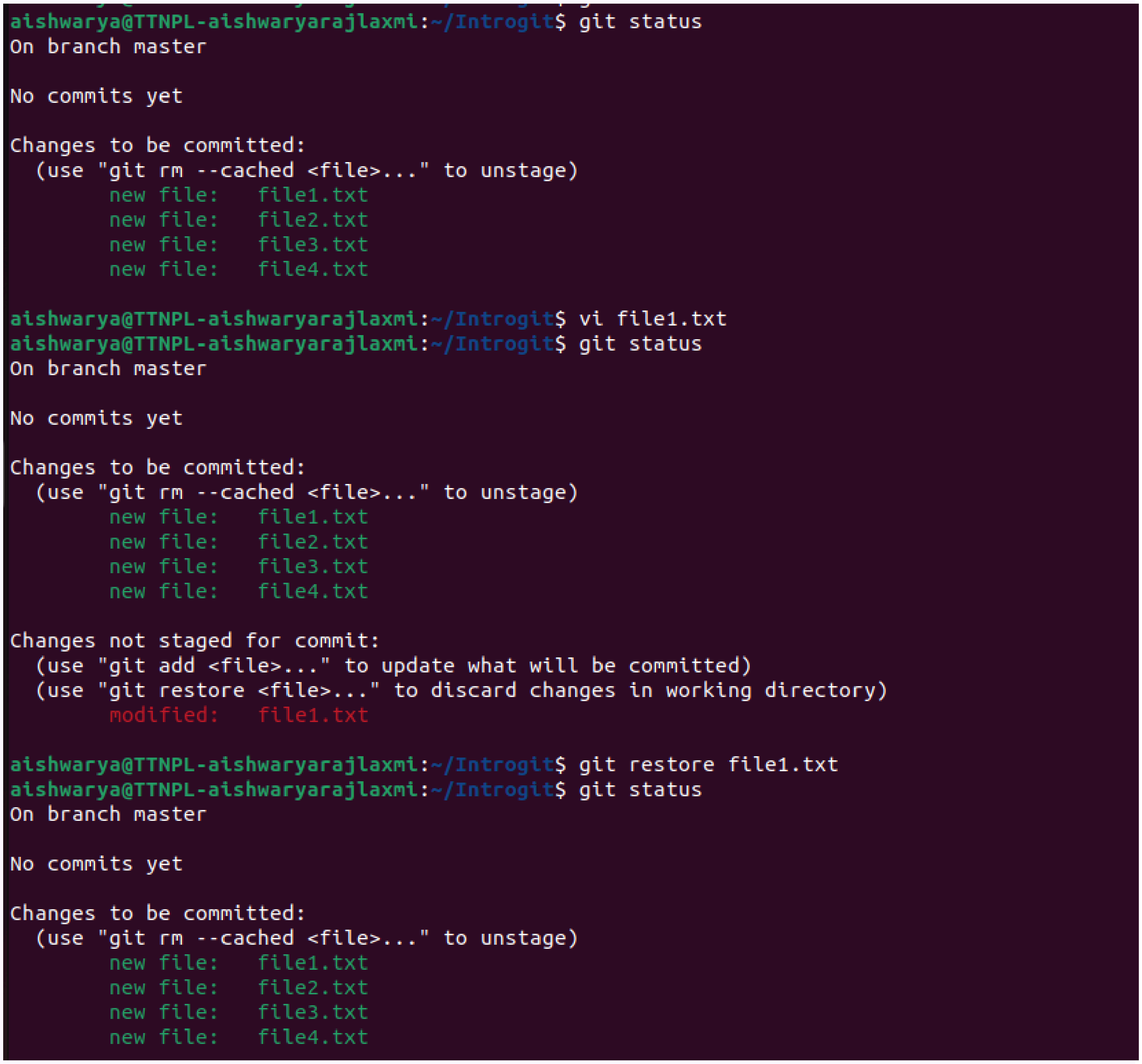


```
aishwarya@TTNPL-aishwaryarajlaxmi:~$ git remote add origin https://github.com/AishwaryaRajLaxmii/IntroductionToGit.git
aishwarya@TTNPL-aishwaryarajlaxmi:~$ git remote -v
origin  https://github.com/AishwaryaRajLaxmii/IntroductionToGit.git (fetch)
origin  https://github.com/AishwaryaRajLaxmii/IntroductionToGit.git (push)
aishwarya@TTNPL-aishwaryarajlaxmi:~$
```

Q7)Undo changes to a particular file

Sol:

Using restore we are able to undo the file sol:

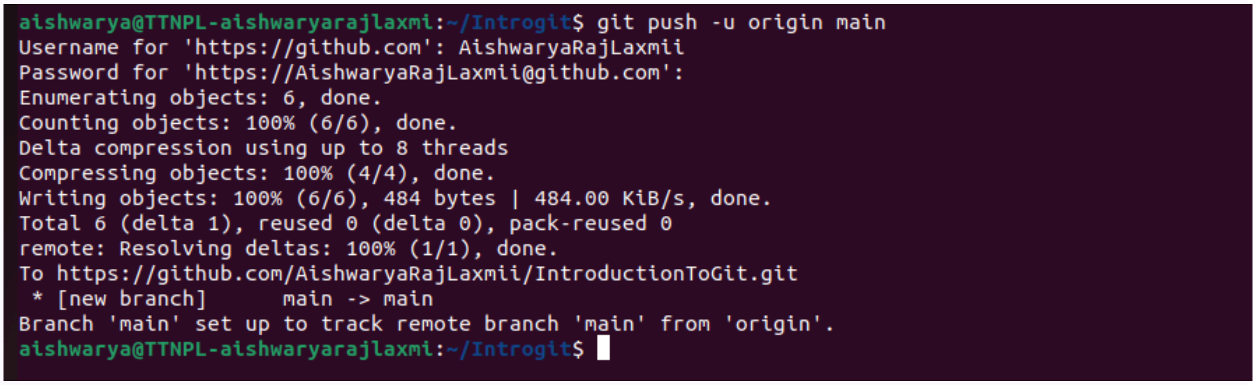


```
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1.txt
        new file:   file2.txt
        new file:   file3.txt
        new file:   file4.txt

aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ vi file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1.txt
        new file:   file2.txt
        new file:   file3.txt
        new file:   file4.txt

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:   file1.txt

aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git restore file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1.txt
        new file:   file2.txt
        new file:   file3.txt
        new file:   file4.txt
```

Q8)Push changes to Github

Sol:

Sending local commits to the remote repository:

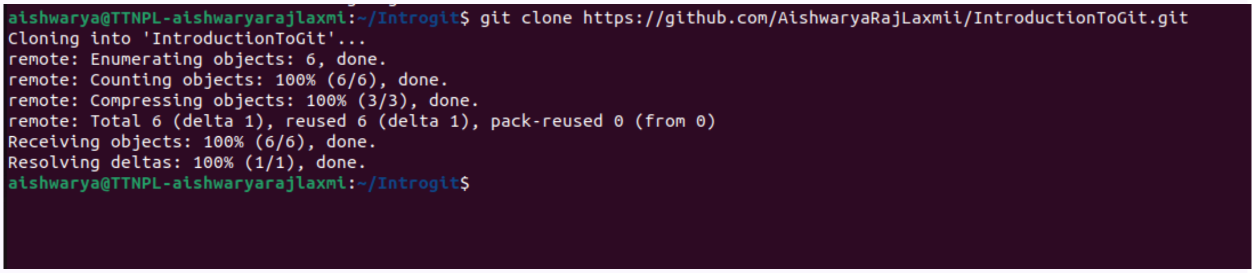


```
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git push -u origin main
Username for 'https://github.com': AishwaryaRajLaxmii
Password for 'https://AishwaryaRajLaxmii@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 484 bytes | 484.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/AishwaryaRajLaxmii/IntroductionToGit.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$
```

Q9)Clone the repository

Sol:

Copy a remote repository to local machine:

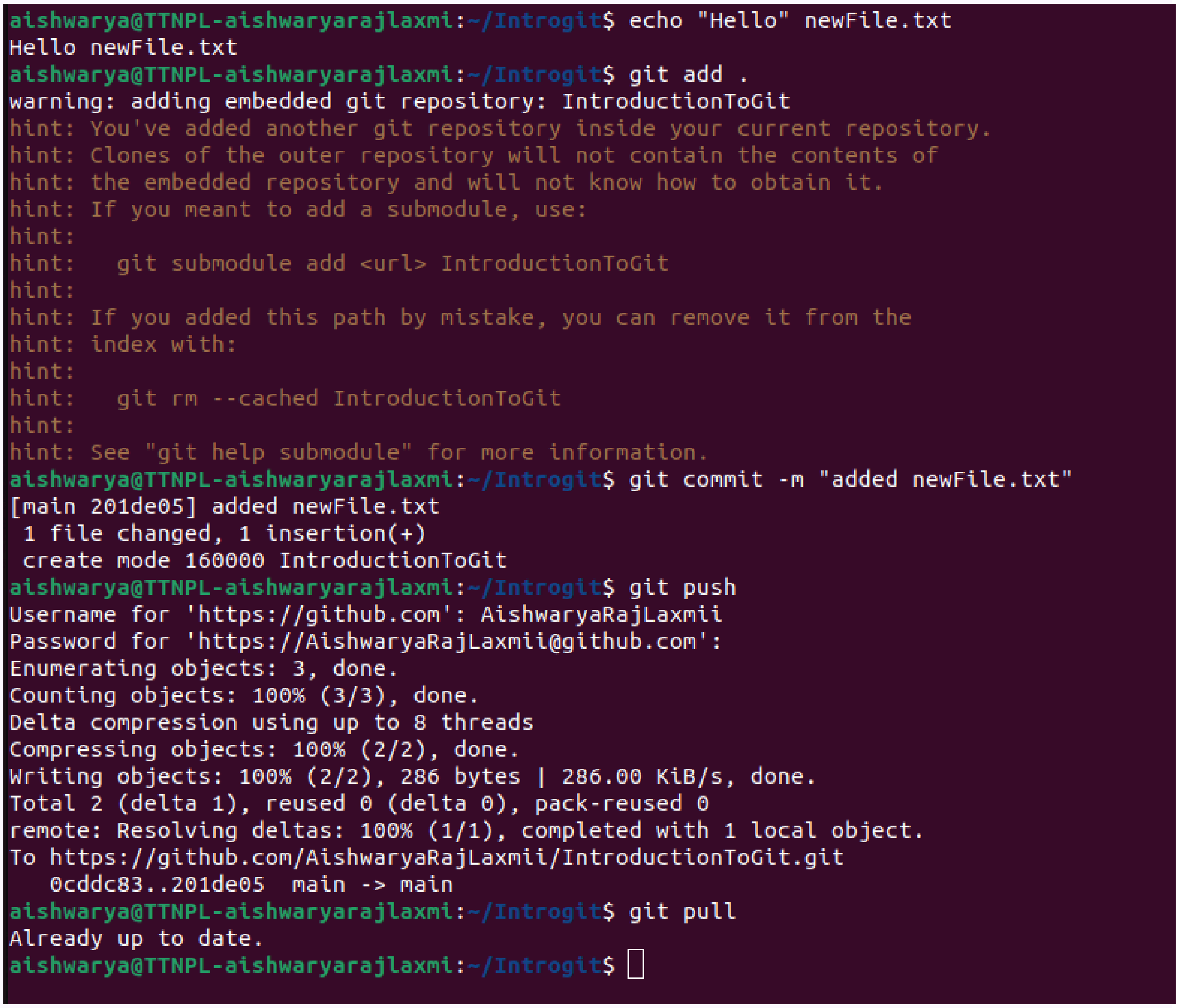


```
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git clone https://github.com/AishwaryaRajLaxmii/IntroductionToGit.git
Cloning into 'IntroductionToGit'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 1), reused 6 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), done.
Resolving deltas: 100% (1/1), done.
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$
```

Q10)Add changes to one of the copies and pull the changes in the other.

Sol:

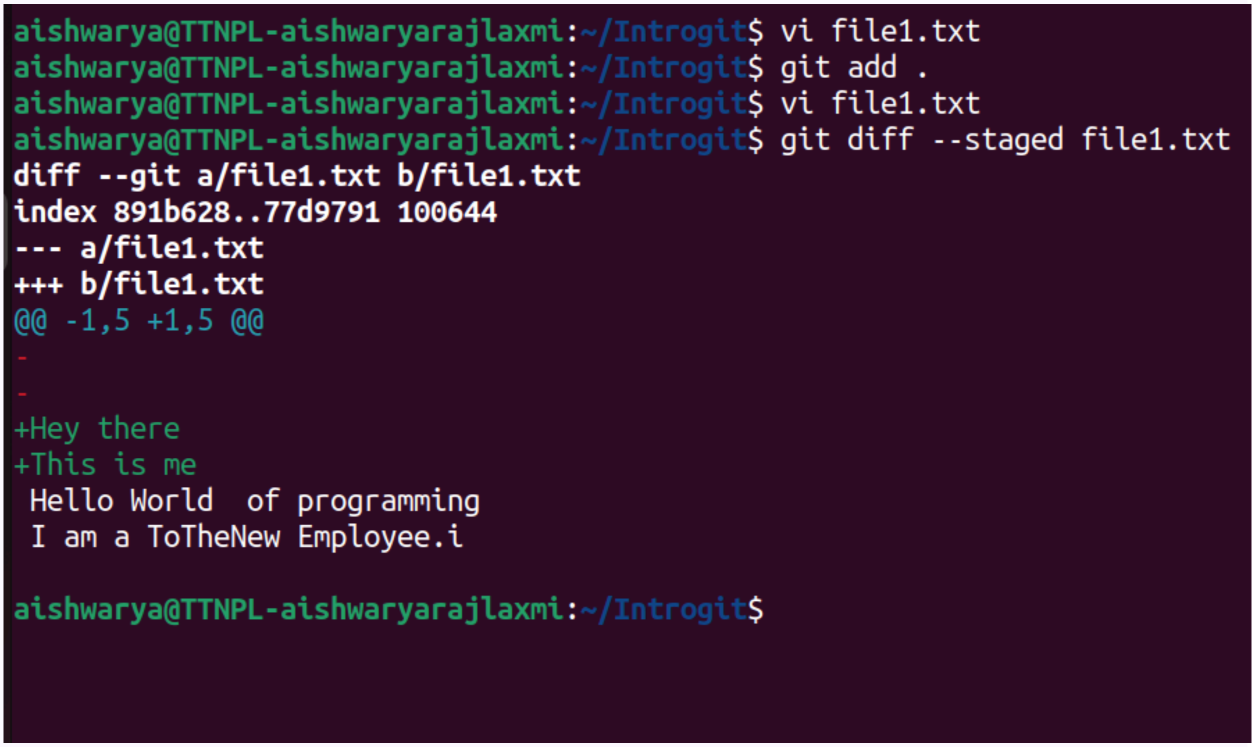
Pull the repository from remote server which means any update in that repository will make our local repository updated



```
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ echo "Hello" newFile.txt
Hello newFile.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git add .
warning: adding embedded git repository: IntroductionToGit
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> IntroductionToGit
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached IntroductionToGit
hint:
hint: See "git help submodule" for more information.
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git commit -m "added newFile.txt"
[main 201de05] added newFile.txt
 1 file changed, 1 insertion(+)
 create mode 160000 IntroductionToGit
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git push
Username for 'https://github.com': AishwaryaRajLaxmii
Password for 'https://AishwaryaRajLaxmii@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 286 bytes | 286.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/AishwaryaRajLaxmii/IntroductionToGit.git
   0cddc83..201de05  main -> main
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git pull
Already up to date.
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$
```

Q11)Check differences between a file and its staged version

sol:

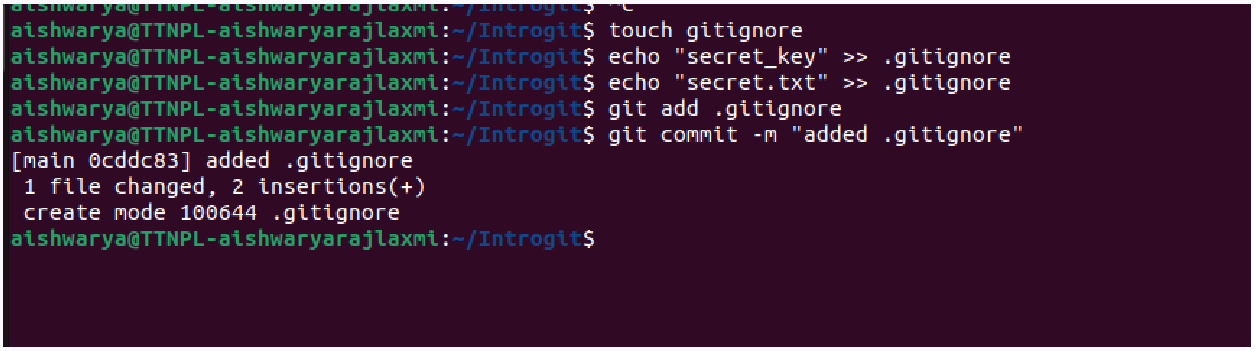


```
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ vi file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git add .
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ vi file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git diff --staged file1.txt
diff --git a/file1.txt b/file1.txt
index 891b628..77d9791 100644
--- a/file1.txt
+++ b/file1.txt
@@ -1,5 +1,5 @@
-
-
+Hey there
+This is me
 Hello World  of programming
 I am a ToTheNew Employee.i

aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$
```

Q12)Ignore a few files to be checked in

Sol:

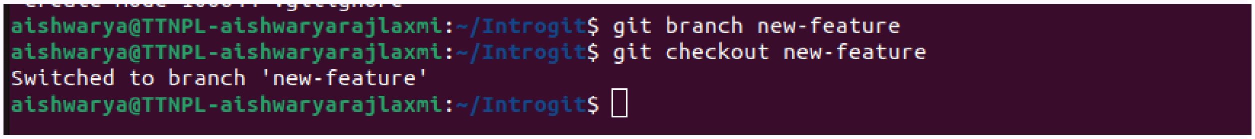
.gitignore does not track the files which we keep inside it.



```
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ ^C
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ touch gitignore
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ echo "secret_key" >> .gitignore
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ echo "secret.txt" >> .gitignore
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git add .gitignore
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git commit -m "added .gitignore"
[main 0cddc83] added .gitignore
 1 file changed, 2 insertions(+)
 create mode 100644 .gitignore
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$
```

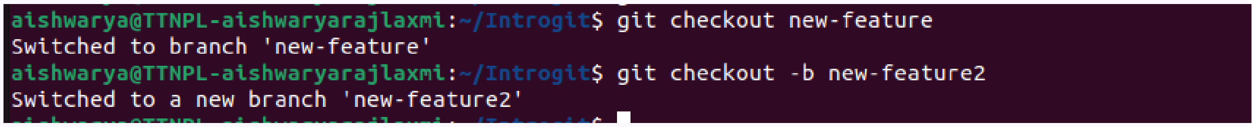
Q13)Create a new branch

sol:



```
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git branch new-feature
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git checkout new-feature
Switched to branch 'new-feature'
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ 
```

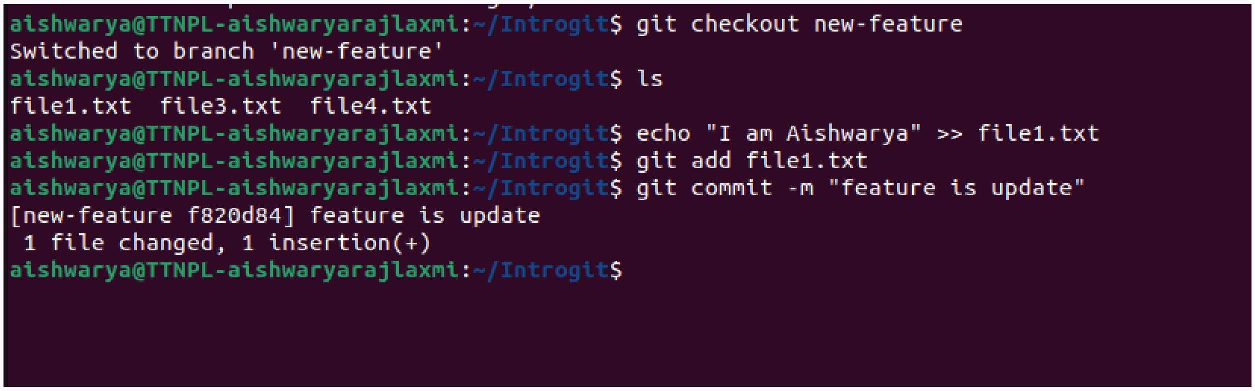
Second way:



```
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git checkout new-feature
Switched to branch 'new-feature'
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git checkout -b new-feature2
Switched to a new branch 'new-feature2'
```

Q14)Diverge them with commits

sol:

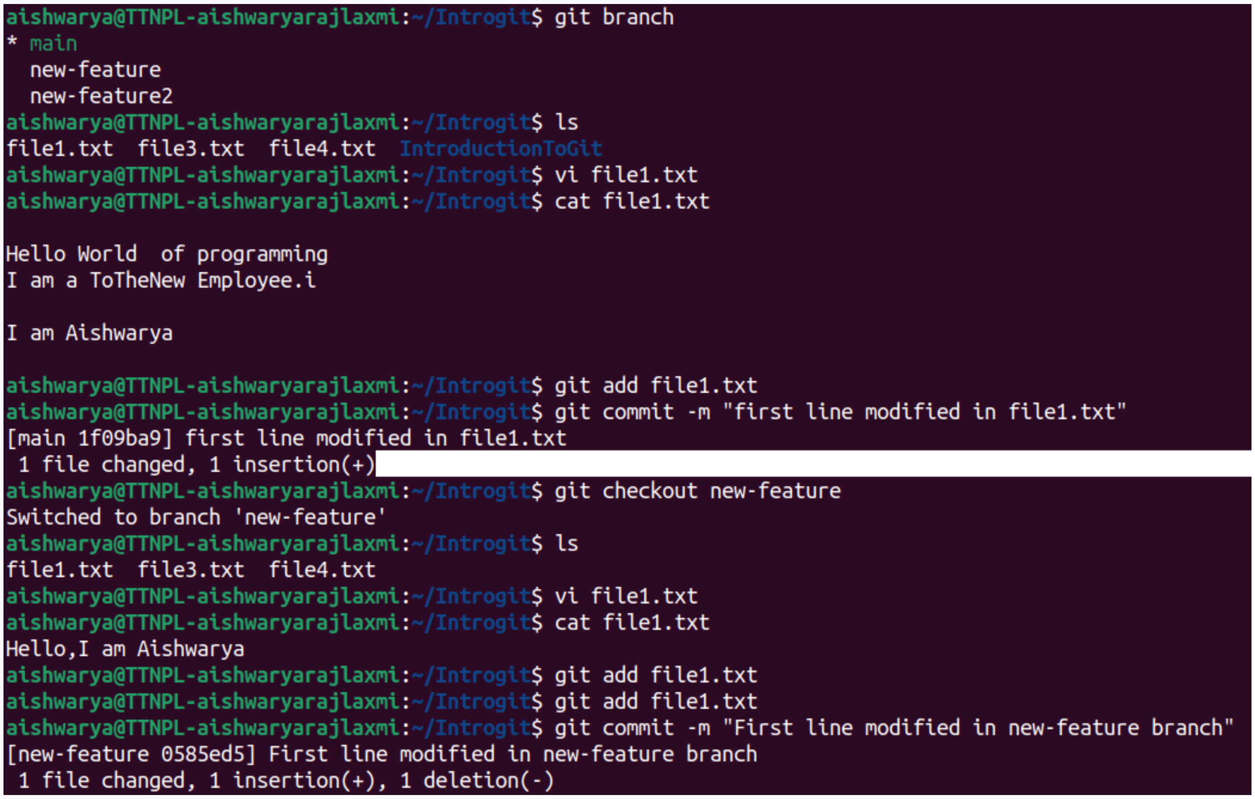


```
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git checkout new-feature
Switched to branch 'new-feature'
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ ls
file1.txt  file3.txt  file4.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ echo "I am Aishwarya" >> file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git add file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git commit -m "feature is update"
[new-feature f820d84] feature is update
 1 file changed, 1 insertion(+)
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$
```

Q15)Edit the same file at the same line on both branches and commit

Sol:

In main branch editing the file1.txt and after that switched to 'new-feature branch' and make changes on same file at same line and committed it.

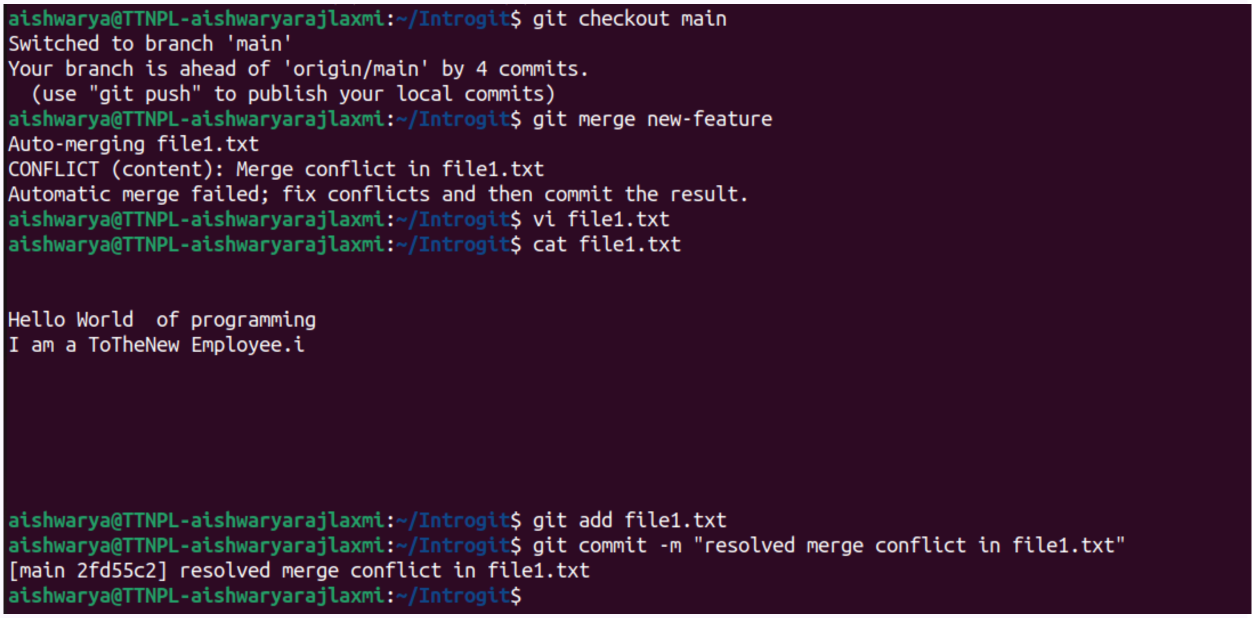


```
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git branch
* main
  new-feature
  new-feature2
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ ls
file1.txt  file3.txt  file4.txt  IntroductionToGit
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ vi file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ cat file1.txt

Hello World  of programming
I am a ToTheNew Employee.i

I am Aishwarya

aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git add file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git commit -m "first line modified in file1.txt"
[main 1f09ba9] first line modified in file1.txt
 1 file changed, 1 insertion(+)
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git checkout new-feature
Switched to branch 'new-feature'
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ ls
file1.txt  file3.txt  file4.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ vi file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ cat file1.txt
Hello,I am Aishwarya
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git add file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git add file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git commit -m "First line modified in new-feature branch"
[new-feature 0585ed5] First line modified in new-feature branch
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Q16)Try merging and resolve merge conflicts

Sol:

Merged the main and new-feature and merge conflict occured so that manually fixed the merged conflict .

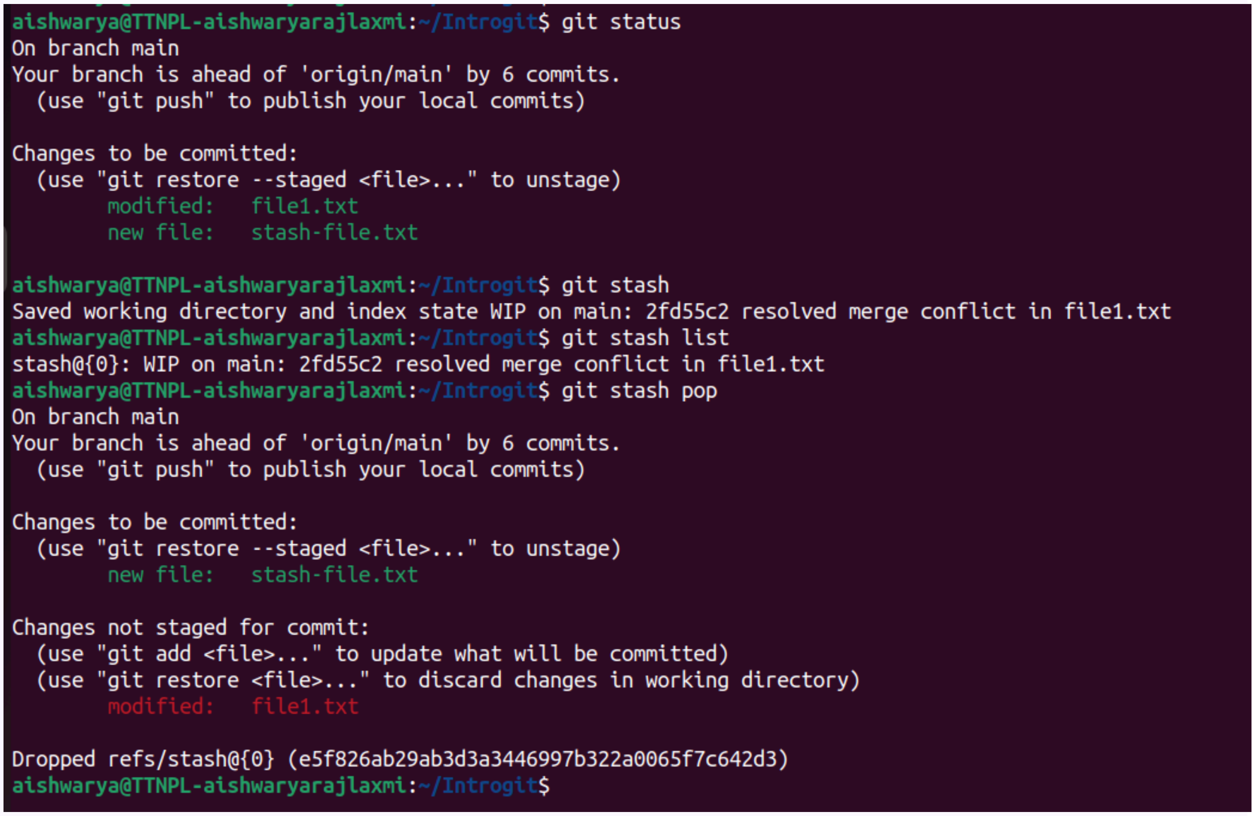


```
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 4 commits.
  (use "git push" to publish your local commits)
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git merge new-feature
Auto-merging file1.txt
CONFLICT (content): Merge conflict in file1.txt
Automatic merge failed; fix conflicts and then commit the result.
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ vi file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ cat file1.txt


Hello World  of programming
I am a ToTheNew Employee.i






aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git add file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git commit -m "resolved merge conflict in file1.txt"
[main 2fd55c2] resolved merge conflict in file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$
```

Q17)Stash the changes and pop them

Sol:

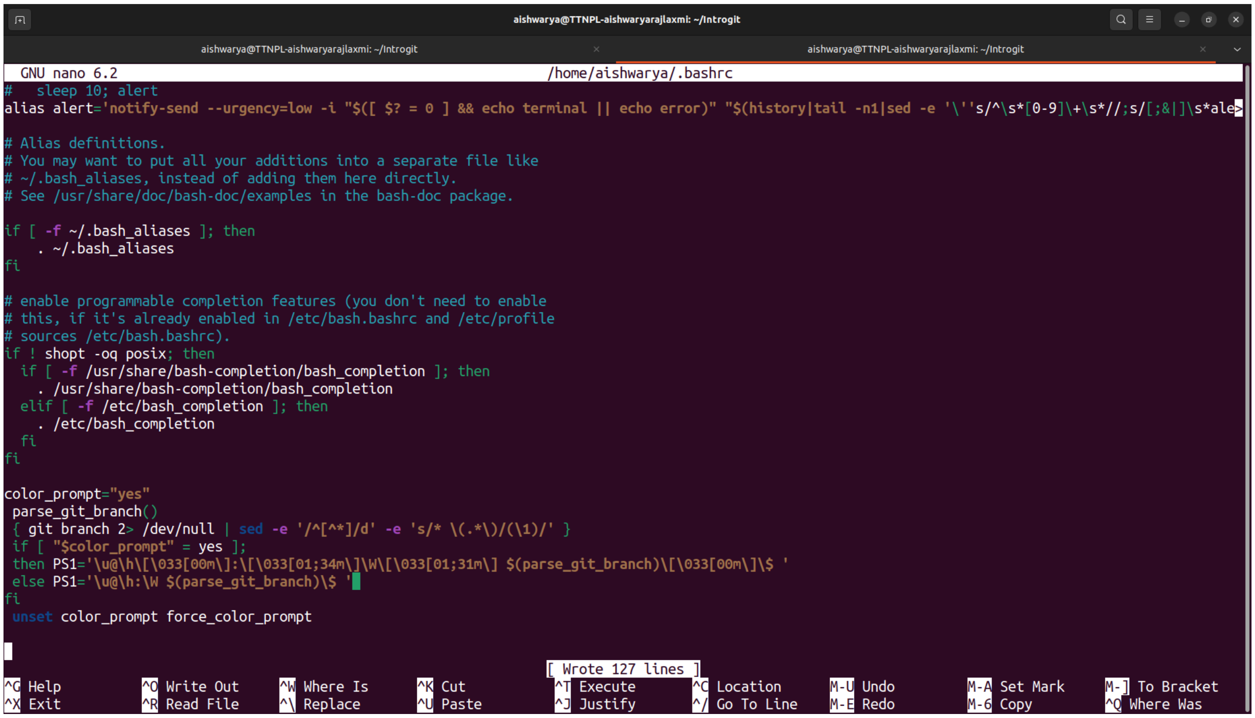
Sometimes, we are working on something but need to switch branches or do something else without committing our changes. we don’t want to lose the work, but it's not ready to be committed so git stash helps us temporarily save our work.



```
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git status
On branch main
Your branch is ahead of 'origin/main' by 6 commits.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   file1.txt
        new file:   stash-file.txt

aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git stash
Saved working directory and index state WIP on main: 2fd55c2 resolved merge conflict in file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git stash list
stash@{0}: WIP on main: 2fd55c2 resolved merge conflict in file1.txt
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$ git stash pop
On branch main
Your branch is ahead of 'origin/main' by 6 commits.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   stash-file.txt

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:   file1.txt

Dropped refs/stash@{0} (e5f826ab29ab3d3a3446997b322a0065f7c642d3)
aishwarya@TTNPL-aishwaryarajlaxmi:~/Introgit$
```

Q18)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

Sol:


aishwarya@TTNPL-aishwaryarajlaxmi: ~/Introgit
aishwarya@TTNPL-aishwaryarajlaxmi: ~/Introgit
aishwarya@TTNPL-aishwaryarajlaxmi: ~/Introgit
GNU nano 6.2
/home/aishwarya/.bashrc

```
#   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*ale>

# 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
```

```
[ Wrote 127 lines ]
```