

Assignment - Introduction to Version Control

Q1)Git Setup

Ans-

1) Installing git and config user

```
akash@TTNPL-akashkumar:~/Desktop$ 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 29 not upgraded.
akash@TTNPL-akashkumar:~/Desktop$ git --version
git version 2.34.1
akash@TTNPL-akashkumar:~/Desktop$ git config --global user.name "Akash Kumar"
git config --global user.email "akash.kumar5@tothenew.com"
akash@TTNPL-akashkumar:~/Desktop$ git config --list
user.name=Akash Kumar
user.email=akash.kumar5@tothenew.com
akash@TTNPL-akashkumar:~/Desktop$
```

2) Generating ssh key

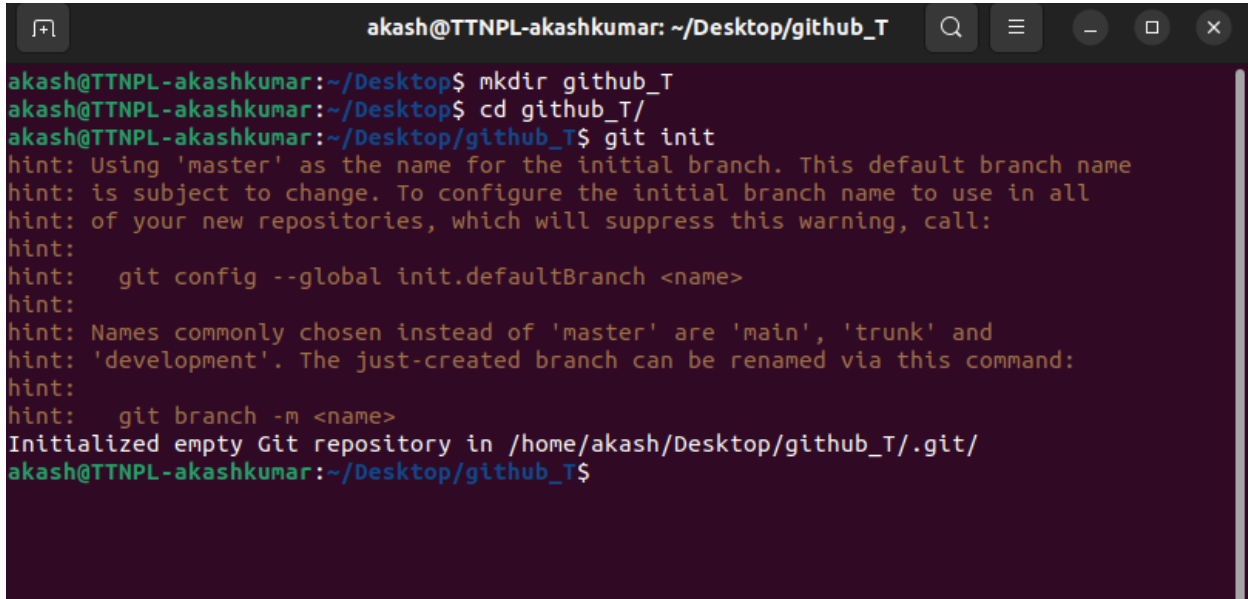
```
akash@TTNPL-akashkumar:~/Desktop$ ssh-keygen -t ed25519 -C "akash.kumar5@tothenew.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/akash/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/akash/.ssh/id_ed25519
Your public key has been saved in /home/akash/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:2EY8HQ6Sj7onkaUWUSc38hpS1oh/0sIIgFNLew366Ig akash.kumar5@tothenew.com
The key's randomart image is:
+--[ED25519 256]--+
|ooo. +B+= .      |
|o.o.+o=0.= .      |
|..o.B.+ = o       |
|  +.B*+.          |
|   Oo+S           |
|  B ..            |
|. + o             |
|E . + .           |
|  o               |
+----[SHA256]-----+
```

3) Steps to add ssh key on github

- Run command `cat ~/.ssh/id_ed25519.pub` and copy result entirely.
- Goto Github SSH setting
- Select New SSH key
- Fill the required fields and save.

Q2)Initialize a Git Repository

Ans -

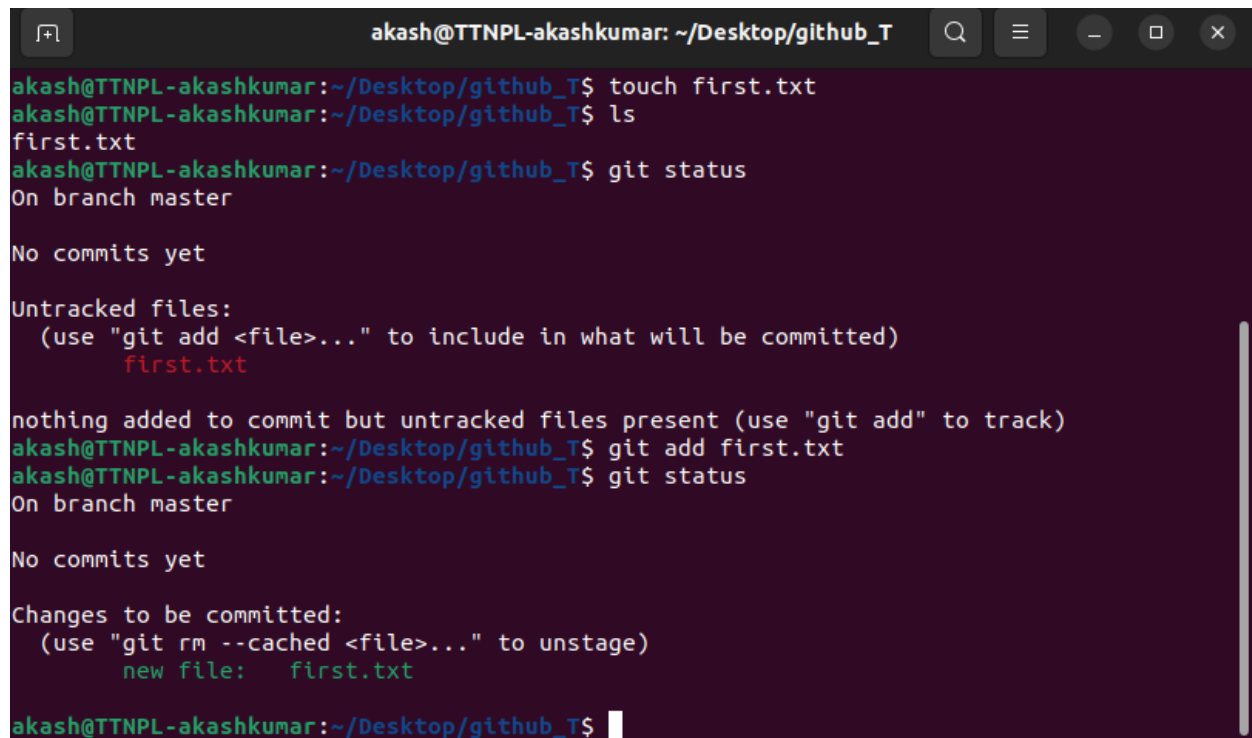
A terminal window with a dark purple background and light green text. The window title is 'akash@TTNPL-akashkumar: ~/Desktop/github_T'. The terminal shows the following commands and output:

```
akash@TTNPL-akashkumar:~/Desktop$ mkdir github_T
akash@TTNPL-akashkumar:~/Desktop$ cd github_T/
akash@TTNPL-akashkumar:~/Desktop/github_T$ 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/akash/Desktop/github_T/.git/
akash@TTNPL-akashkumar:~/Desktop/github_T$
```

Following are the Steps -

- 1) Make a Demo Directory
- 2) Move to that directory
- 3) Use `git init` - to initialize git

Q3)Add files to the repository

A terminal window with a dark purple background and light green text. The window title is 'akash@TTNPL-akashkumar: ~/Desktop/github_T'. The terminal shows the following sequence of commands and outputs: 1. 'touch first.txt' is executed. 2. 'ls' is executed, showing 'first.txt'. 3. 'git status' is executed, showing 'On branch master' and 'No commits yet'. 4. 'git status' is executed again, showing 'Untracked files: (use "git add <file>..." to include in what will be committed)' followed by 'first.txt' in red. 5. A message 'nothing added to commit but untracked files present (use "git add" to track)' is displayed. 6. 'git add first.txt' is executed. 7. 'git status' is executed, showing 'On branch master' and 'No commits yet'. 8. 'git status' is executed again, showing 'Changes to be committed: (use "git rm --cached <file>..." to unstage)' followed by 'new file: first.txt' in green. 9. The prompt 'akash@TTNPL-akashkumar:~/Desktop/github_T\$' is shown at the bottom with a cursor.

```
akash@TTNPL-akashkumar:~/Desktop/github_T$ touch first.txt
akash@TTNPL-akashkumar:~/Desktop/github_T$ ls
first.txt
akash@TTNPL-akashkumar:~/Desktop/github_T$ git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
akash@TTNPL-akashkumar:~/Desktop/github_T$ git add first.txt
akash@TTNPL-akashkumar:~/Desktop/github_T$ git status
On branch master

No commits yet

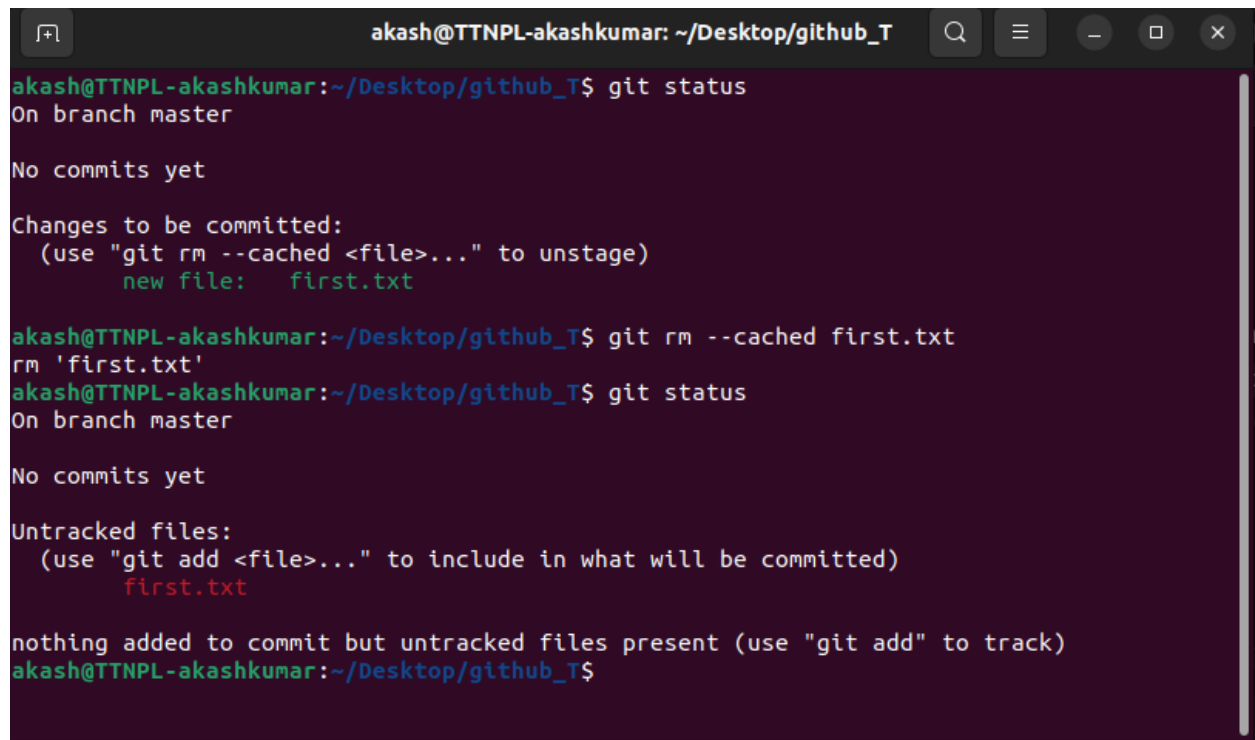
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
       new file:   first.txt

akash@TTNPL-akashkumar:~/Desktop/github_T$
```

Following are the Steps -

- First Add a file using touch
- Then check git status for current status of the file
- Then if file is not tracked use git add <filename> or git add . to add the file
- Then check if the status is changed or not

Q4)Unstage 1 file

A terminal window with a dark purple background and white text. The window title is 'akash@TTNPL-akashkumar: ~/Desktop/github_T'. The terminal shows the following commands and output:

```
akash@TTNPL-akashkumar:~/Desktop/github_T$ git status
On branch master

No commits yet

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

akash@TTNPL-akashkumar:~/Desktop/github_T$ git rm --cached first.txt
rm 'first.txt'
akash@TTNPL-akashkumar:~/Desktop/github_T$ git status
On branch master

No commits yet

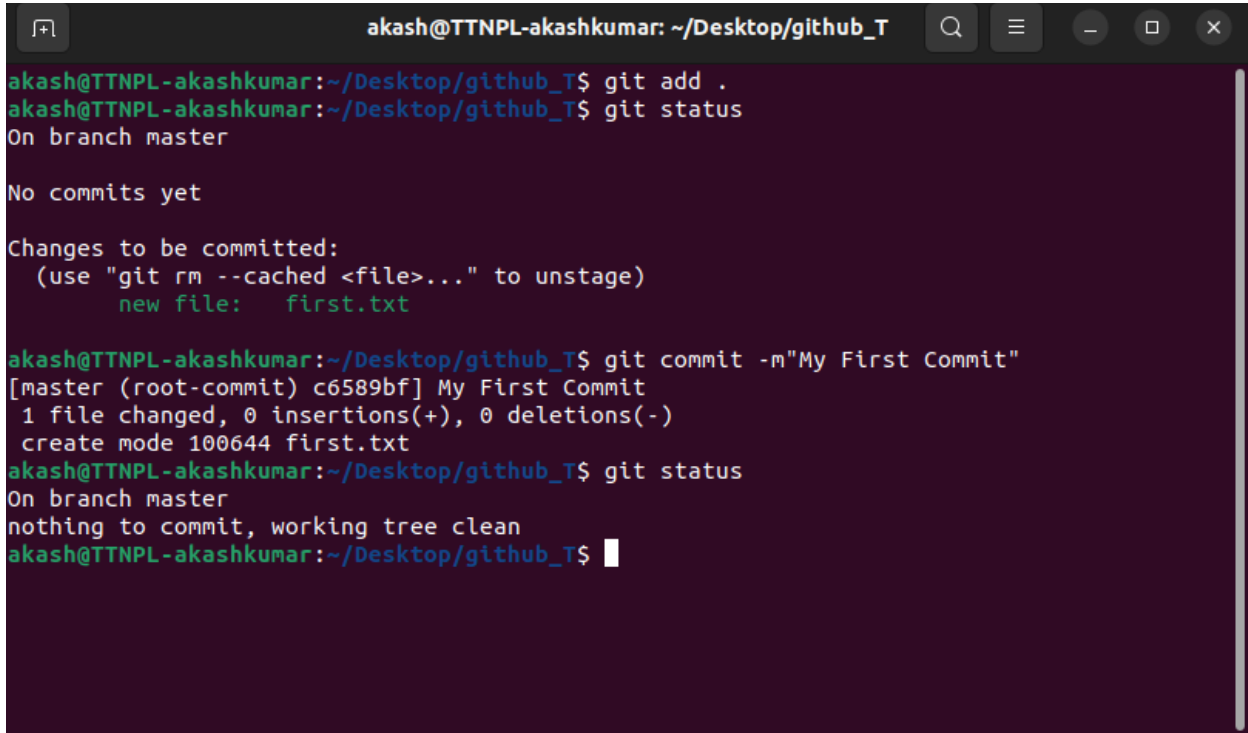
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        first.txt

nothing added to commit but untracked files present (use "git add" to track)
akash@TTNPL-akashkumar:~/Desktop/github_T$
```

Following are the Steps -

- Use command `git rm --cached <filename>` to basically unstage it without removing from actual directory

Q5)Commit the file

A terminal window with a dark purple background. The title bar shows 'akash@TTNPL-akashkumar: ~/Desktop/github_T'. The terminal text is as follows:

```
akash@TTNPL-akashkumar:~/Desktop/github_T$ git add .
akash@TTNPL-akashkumar:~/Desktop/github_T$ git status
On branch master

No commits yet

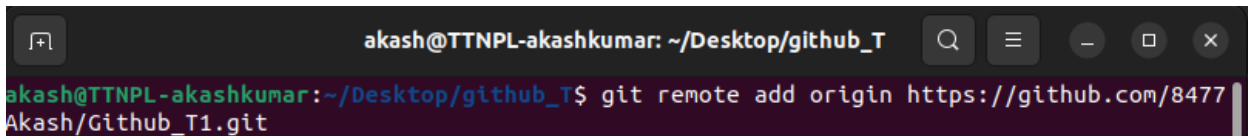
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   first.txt

akash@TTNPL-akashkumar:~/Desktop/github_T$ git commit -m"My First Commit"
[master (root-commit) c6589bf] My First Commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 first.txt
akash@TTNPL-akashkumar:~/Desktop/github_T$ git status
On branch master
nothing to commit, working tree clean
akash@TTNPL-akashkumar:~/Desktop/github_T$
```

Following are the Steps -

- Add all the relevant files you want to commit
- Use command `git commit -m"your message"`

Q6)Add a remote

A terminal window with a dark purple background. The title bar shows 'akash@TTNPL-akashkumar: ~/Desktop/github_T'. The terminal text is as follows:

```
akash@TTNPL-akashkumar:~/Desktop/github_T$ git remote add origin https://github.com/8477
Akash/Github_T1.git
```

- Use command `git remote <url>`

Q7)Undo changes to a particular file

```
akash@TTNPL-akashkumar:~/Desktop$ cd github_T/
akash@TTNPL-akashkumar:~/Desktop/github_T$ vim first.txt
akash@TTNPL-akashkumar:~/Desktop/github_T$ git status
On branch main
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:   first.txt

no changes added to commit (use "git add" and/or "git commit -a")
akash@TTNPL-akashkumar:~/Desktop/github_T$ git restore first.txt
akash@TTNPL-akashkumar:~/Desktop/github_T$ git status
On branch main
nothing to commit, working tree clean
akash@TTNPL-akashkumar:~/Desktop/github_T$
```

- Use command git restore <filename>

Q8)Push changes to Github

```
akash@TTNPL-akashkumar:~/Desktop/github_T$ git remote -v
origin https://github.com/8477Akash/Github_T1.git (fetch)
origin https://github.com/8477Akash/Github_T1.git (push)
akash@TTNPL-akashkumar:~/Desktop/github_T$ git remote set-url origin git@github.
com:8477Akash/Github_T1.git
akash@TTNPL-akashkumar:~/Desktop/github_T$ git remote -v
origin git@github.com:8477Akash/Github_T1.git (fetch)
origin git@github.com:8477Akash/Github_T1.git (push)
akash@TTNPL-akashkumar:~/Desktop/github_T$ git branch -M main
akash@TTNPL-akashkumar:~/Desktop/github_T$ git push -u origin main
Please type 'yes' to continue the setup process, or 'no' to cancel
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 222 bytes | 222.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:8477Akash/Github_T1.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
akash@TTNPL-akashkumar:~/Desktop/github_T$
```

Following are the Steps -

- Use git remote -v to check remote directory
- Use git push -u origin main

Q9)Clone the repository

```
akash@TTNPL-akashkumar: ~/Desktop/demo
akash@TTNPL-akashkumar:~/Desktop/demo$ ls
akash@TTNPL-akashkumar:~/Desktop/demo$ 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/akash/Desktop/demo/.git/
akash@TTNPL-akashkumar:~/Desktop/demo$ git clone git@github.com:8477Akash/Github
_T1.git
Cloning into 'Github_T1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
akash@TTNPL-akashkumar:~/Desktop/demo$ ls
Github_T1
akash@TTNPL-akashkumar:~/Desktop/demo$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Github_T1/

nothing added to commit but untracked files present (use "git add" to track)
akash@TTNPL-akashkumar:~/Desktop/demo$
```

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

```
akash@TTNPL-akashkumar: ~/Desktop/demo/Github_T1
nothing added to commit but untracked files present (use "git add" to track)
akash@TTNPL-akashkumar:~/Desktop/demo$ cd ..
akash@TTNPL-akashkumar:~/Desktop$ cd github_T/
akash@TTNPL-akashkumar:~/Desktop/github_T$ vim first.txt
akash@TTNPL-akashkumar:~/Desktop/github_T$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   first.txt

no changes added to commit (use "git add" and/or "git commit -a")
akash@TTNPL-akashkumar:~/Desktop/github_T$ git add .
akash@TTNPL-akashkumar:~/Desktop/github_T$ git commit -m"Updated the First file"
[main 9e61637] Updated the First file
 1 file changed, 2 insertions(+)
akash@TTNPL-akashkumar:~/Desktop/github_T$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 282 bytes | 282.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:8477Akash/Github_T1.git
   c6589bf..9e61637  main -> main
akash@TTNPL-akashkumar:~/Desktop/github_T$ cd ..
akash@TTNPL-akashkumar:~/Desktop$ cd demo/
```

Following are the Steps -

- Modify files from first copy and push changes to github
- Now move to next copy go to the cloned repository and use command
- git pull origin main - to pull

```
akash@TTNPL-akashkumar:~/Desktop/demo$ cd Github_T1/
akash@TTNPL-akashkumar:~/Desktop/demo/Github_T1$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 262 bytes | 262.00 KiB/s, done.
From github.com:8477Akash/Github_T1
 * branch          main      -> FETCH_HEAD
   c6589bf..9e61637  main      -> origin/main
Updating c6589bf..9e61637
Fast-forward
 first.txt | 2 ++
 1 file changed, 2 insertions(+)
akash@TTNPL-akashkumar:~/Desktop/demo/Github_T1$
```


Q11) Check differences between a file and its staged version

```
akash@TTNPL-akashkumar:~/Desktop/github_T$ vim first.txt
akash@TTNPL-akashkumar:~/Desktop/github_T$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

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:   first.txt

no changes added to commit (use "git add" and/or "git commit -a")
akash@TTNPL-akashkumar:~/Desktop/github_T$ git diff first.txt
diff --git a/first.txt b/first.txt
index 184b5fb..dd67dbe 100644
--- a/first.txt
+++ b/first.txt
@@ -1,4 +1,7 @@
  this is the modified file

  -new modification
  +new modificatio
  +
  +
  +fdhfsjjfdkdkn
akash@TTNPL-akashkumar:~/Desktop/github_T$
```

Q12)Ignore a few files to be checked in

```
akash@TTNPL-akashkumar:~/Desktop/github_T$ touch 1.txt 2.txt
akash@TTNPL-akashkumar:~/Desktop/github_T$ ls
1.txt 2.txt first.txt
akash@TTNPL-akashkumar:~/Desktop/github_T$ touch .gitignore
akash@TTNPL-akashkumar:~/Desktop/github_T$ vim .gitignore
akash@TTNPL-akashkumar:~/Desktop/github_T$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

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:   first.txt

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

no changes added to commit (use "git add" and/or "git commit -a")
akash@TTNPL-akashkumar:~/Desktop/github_T$
```

Following are the Steps -

- Create new files along with .gitignore
- Add file names to git ignore

Even though we have new files they are not showed in untracked or modified files

Q13)Create a new branch

```
akash@TTNPL-akashkumar:~/Desktop/github_T$ git branch Main2
akash@TTNPL-akashkumar:~/Desktop/github_T$ git checkout Main2
M       first.txt
Switched to branch 'Main2'
akash@TTNPL-akashkumar:~/Desktop/github_T$
```

Q14)Diverge them with commits

```
akash@TTNPL-akashkumar:~/Desktop/github_T$ git add .
akash@TTNPL-akashkumar:~/Desktop/github_T$ git status
On branch Main2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore
    new file:   a1.txt
    modified:   first.txt

akash@TTNPL-akashkumar:~/Desktop/github_T$ git push origin Main2
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 286 bytes | 286.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'Main2' on GitHub by visiting:
remote:   https://github.com/8477Akash/Github_T1/pull/new/Main2
remote:
To github.com:8477Akash/Github_T1.git
 * [new branch]      Main2 -> Main2
akash@TTNPL-akashkumar:~/Desktop/github_T$
```

LOG OF BRANCHES

```
[new branch]      Main2 -> Main2
akash@TTNPL-akashkumar:~/Desktop/github_T$ git log
commit a53be730d56581ed0e9ed0e1e5011b40db068c10 (HEAD -> Main2, origin/Main2, main)
Author: Akash Kumar <akash.kumar5@tothenew.com>
Date:   Thu Jun 12 14:13:40 2025 +0530

    modified file

commit 9e61637125cfde3d2eacec95b184b21e7966a273 (origin/main)
Author: Akash Kumar <akash.kumar5@tothenew.com>
Date:   Thu Jun 12 13:51:58 2025 +0530

    Updated the First file

commit c6589bf61b29144b6f4c5e34a0d22590d7afa96d
Author: Akash Kumar <akash.kumar5@tothenew.com>
Date:   Thu Jun 12 12:37:19 2025 +0530

    My First Commit
akash@TTNPL-akashkumar:~/Desktop/github_T$
```

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

Branch Main2

```

akash@TTNPL-akashkumar:~/Desktop$ cd github_T/
akash@TTNPL-akashkumar:~/Desktop/github_T$ git branch
* Main2
  main
akash@TTNPL-akashkumar:~/Desktop/github_T$ git status
On branch Main2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   a1.txt

akash@TTNPL-akashkumar:~/Desktop/github_T$ git commit -m "New file"
[Main2 939425b] New file
 1 file changed, 1 insertion(+), 1 deletion(-)
akash@TTNPL-akashkumar:~/Desktop/github_T$ git push origin Main2
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 316 bytes | 316.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:8477Akash/Github_T1.git
   77ba7b5..939425b  Main2 -> Main2

```

Branch main

```

akash@TTNPL-akashkumar:~/Desktop/github_T$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   a1.txt

no changes added to commit (use "git add" and/or "git commit -a")
akash@TTNPL-akashkumar:~/Desktop/github_T$ git add .
akash@TTNPL-akashkumar:~/Desktop/github_T$ git commit -m "Edited"
[main 9a9f165] Edited
 1 file changed, 1 insertion(+)
akash@TTNPL-akashkumar:~/Desktop/github_T$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 314 bytes | 314.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:8477Akash/Github_T1.git
   876eddf..9a9f165  main -> main

```

Q16)Try merging and resolve merge conflicts

```
akash@TTNPL-akashkumar:~/Desktop/github_T$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
akash@TTNPL-akashkumar:~/Desktop/github_T$ git merge Main2
Merge made by the 'ort' strategy.
akash@TTNPL-akashkumar:~/Desktop/github_T$ git push origin main
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 231 bytes | 231.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:8477Akash/Github_T1.git
 6b298a4..c214ef1  main -> main
akash@TTNPL-akashkumar:~/Desktop/github_T$
```

Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



Q17)Stash the changes and pop them

```
00290847: c214ef1 main -> main
akash@TTNPL-akashkumar:~/Desktop/github_T$ git stash
No local changes to save
akash@TTNPL-akashkumar:~/Desktop/github_T$ vim a1.txt
akash@TTNPL-akashkumar:~/Desktop/github_T$ git stash
Saved working directory and index state WIP on main: c214ef1 Merging
akash@TTNPL-akashkumar:~/Desktop/github_T$ git stash pop
On branch main
Your branch is up to date with 'origin/main'.

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:   a1.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (bb57160c8730b0963a827d7d30f6ec9e776d5cc0)
akash@TTNPL-akashkumar:~/Desktop/github_T$
```

- Make some changes to be stashed
- Use git stash to stash the changes
- Use git stash pop to pop them

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
```

```

+
akash@TTNPL-akashkumar: ~/Desktop
akash@TTNPL-akashkumar:~/Desktop$ vim ~/.bashrc
akash@TTNPL-akashkumar:~/Desktop$ source ~/.bashrc
akash@TTNPL-akashkumar:Desktop $ cd github_T/
akash@TTNPL-akashkumar:github_T (main)$ vim ~/.bashrc
```