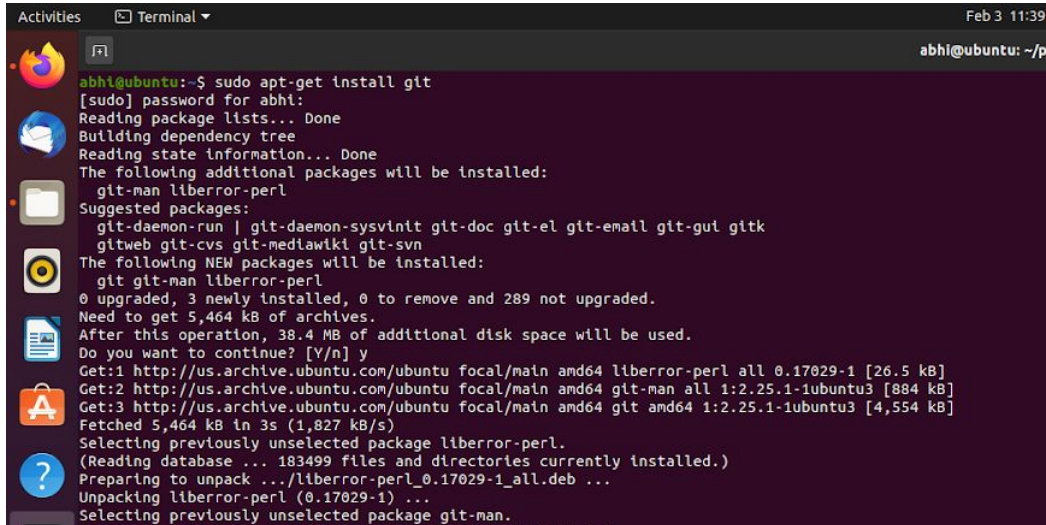


1. Git Setup

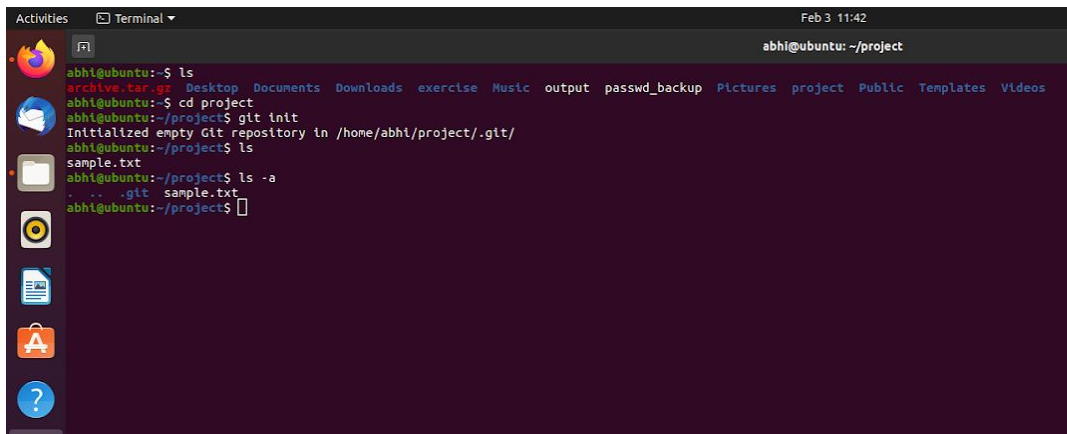
Command: `sudo apt-get install git`

A terminal window titled 'Terminal' with a dark background. The user 'abhi@ubuntu' is in the directory '~'. The command 'sudo apt-get install git' is entered. The terminal shows the password prompt, package lists, dependency tree, and state information. It lists additional packages to be installed: git-man, liberror-perl, and suggested packages: git-daemon-run, git-daemon-sysvinit, git-doc, git-el, git-email, git-gui, gitk, gitweb, git-cvs, git-mediawiki, git-svn. It states that 0 packages are upgraded, 3 are newly installed, and 289 are not upgraded. The total size of archives to be downloaded is 5,464 kB, and an additional 38.4 MB of disk space will be used. The user confirms the installation with 'y'. The terminal shows the download progress for liberror-perl and git-man, and the unpacking process for both packages.

```
abhi@ubuntu:~$ sudo apt-get install git
[sudo] password for abhi:
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 289 not upgraded.
Need to get 5,464 kB of archives.
After this operation, 38.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu focal/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal/main amd64 git-man all 1:2.25.1-1ubuntu3 [884 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal/main amd64 git amd64 1:2.25.1-1ubuntu3 [4,554 kB]
Fetched 5,464 kB in 3s (1,827 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 183499 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
Selecting previously unselected package git-man.
(Reading database ... 183499 files and directories currently installed.)
Preparing to unpack .../git-man_1:2.25.1-1ubuntu3_all.deb ...
Unpacking git-man (1:2.25.1-1ubuntu3) ...
```

2. Initialize a Git Repository

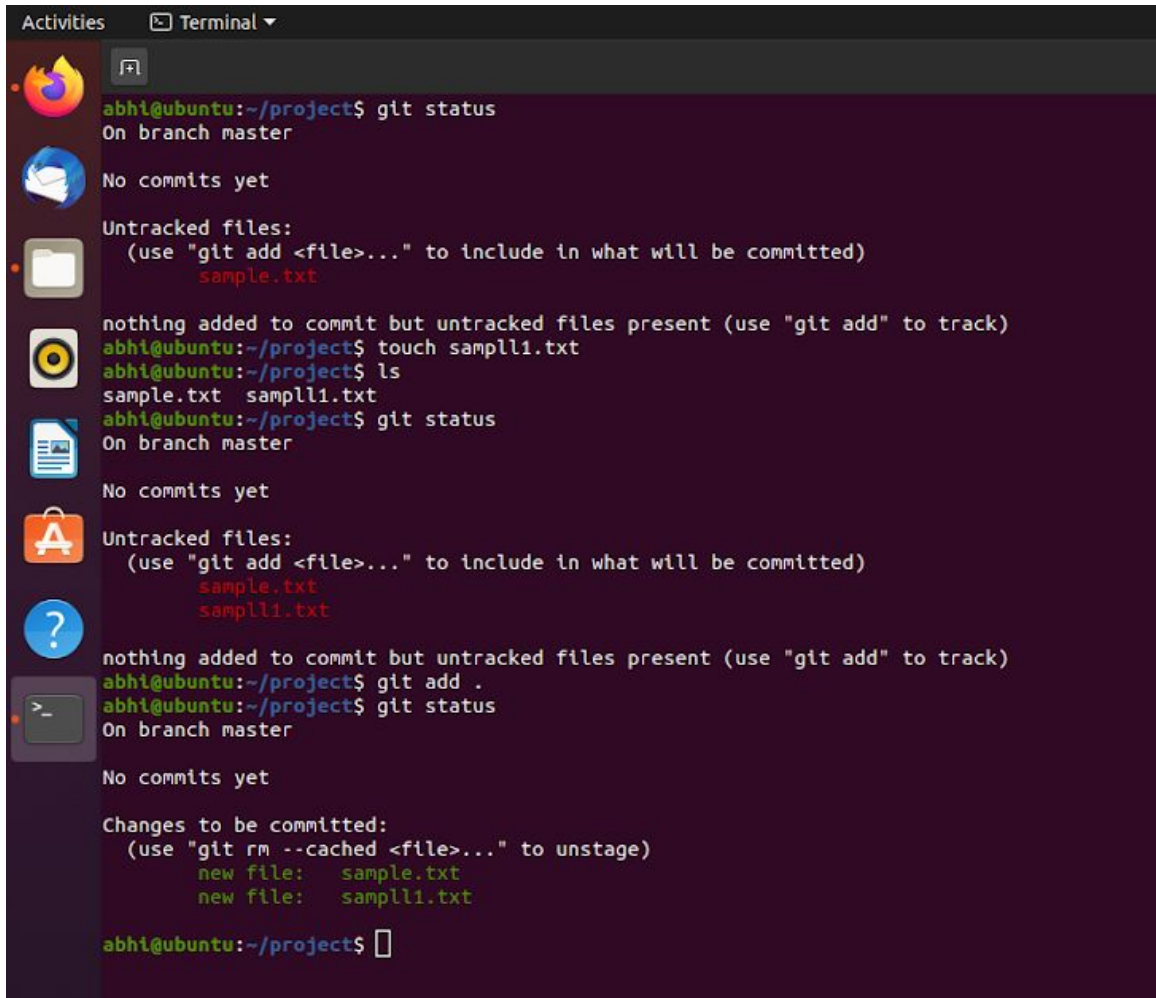
Command: `git init`

A terminal window titled 'Terminal' with a dark background. The user 'abhi@ubuntu' is in the directory '~/project'. The command 'git init' is entered. The terminal shows the message 'Initialized empty Git repository in /home/abhi/project/.git/'. The user then runs 'ls' and 'ls -a' to show the contents of the directory, which include 'sample.txt' and the newly created '.git' directory.

```
abhi@ubuntu:~/project$ git init
Initialized empty Git repository in /home/abhi/project/.git/
abhi@ubuntu:~/project$ ls
sample.txt
abhi@ubuntu:~/project$ ls -a
.  ..  .git  sample.txt
abhi@ubuntu:~/project$
```

3. Add files to the repository

Command: `git add .`

A screenshot of a Linux terminal window with a dark purple background. The window title is "Activities" and "Terminal". On the left side, there is a vertical dock with icons for Firefox, a mail client, a file manager, a media player, a document viewer, an application store, a help icon, and a terminal icon. The terminal shows the following sequence of commands and output:

```
abhi@ubuntu:~/project$ git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
abhi@ubuntu:~/project$ touch samplli.txt
abhi@ubuntu:~/project$ ls
sample.txt  samplli.txt
abhi@ubuntu:~/project$ git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
abhi@ubuntu:~/project$ git add .
abhi@ubuntu:~/project$ git status
On branch master

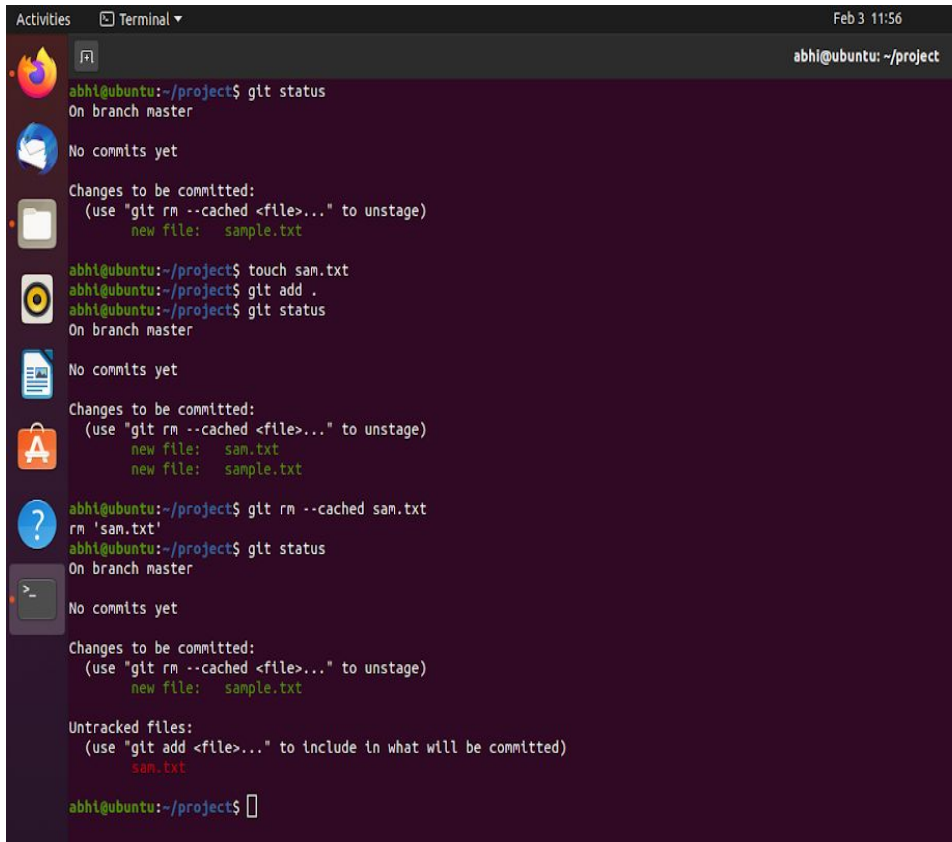
No commits yet

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

abhi@ubuntu:~/project$
```

4. Unstage 1 file

Command: `git rm -f sample1.txt`

A terminal window titled 'Terminal' with a date and time of 'Feb 3 11:56' and a user location of 'abhi@ubuntu: ~/project'. The terminal shows a sequence of commands and their outputs. First, 'git status' is run, showing 'On branch master', 'No commits yet', and 'Changes to be committed: (use "git rm --cached <file>..." to unstage) new file: sample.txt'. Then, 'touch sam.txt' is run. Next, 'git add .' is run. Then, 'git status' is run again, showing 'On branch master', 'No commits yet', and 'Changes to be committed: (use "git rm --cached <file>..." to unstage) new file: sam.txt new file: sample.txt'. Then, 'git rm --cached sam.txt' is run, followed by 'rm 'sam.txt''. Finally, 'git status' is run, showing 'On branch master', 'No commits yet', 'Changes to be committed: (use "git rm --cached <file>..." to unstage) new file: sample.txt', and 'Untracked files: (use "git add <file>..." to include in what will be committed) sam.txt'. The prompt 'abhi@ubuntu:~/project\$' is shown at the bottom.

```
abhi@ubuntu:~/project$ git status
On branch master

No commits yet

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

abhi@ubuntu:~/project$ touch sam.txt
abhi@ubuntu:~/project$ git add .
abhi@ubuntu:~/project$ git status
On branch master

No commits yet

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

abhi@ubuntu:~/project$ git rm --cached sam.txt
rm 'sam.txt'
abhi@ubuntu:~/project$ git status
On branch master

No commits yet

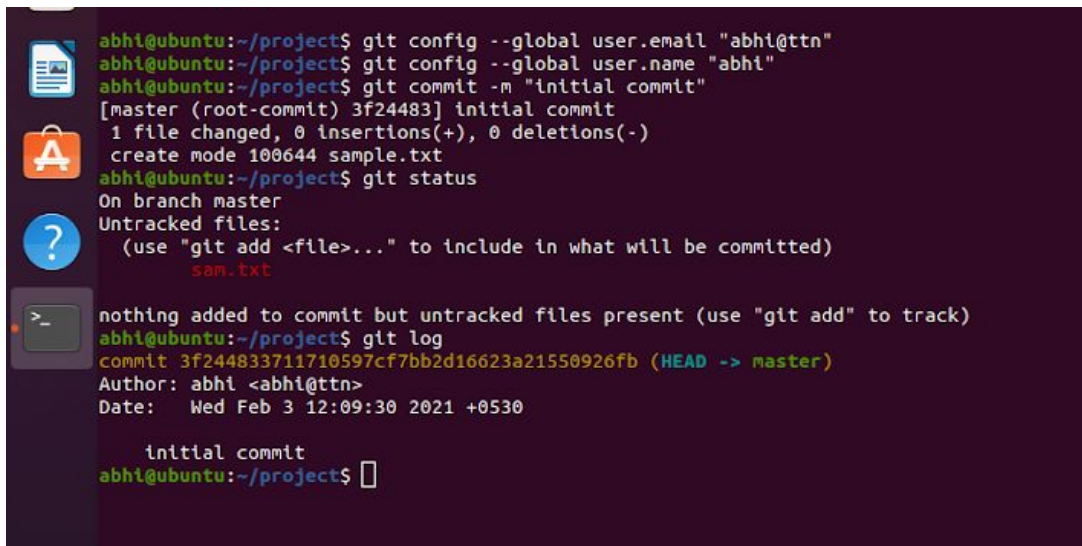
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   sample.txt

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

abhi@ubuntu:~/project$
```

5. Commit the file

Command:

A terminal window showing a sequence of git commands and their outputs. First, 'git config --global user.email "abhi@ttn"' is run. Then, 'git config --global user.name "abhi"' is run. Then, 'git commit -m "initial commit"' is run, showing '[master (root-commit) 3f24483] initial commit' and '1 file changed, 0 insertions(+), 0 deletions(-) create mode 100644 sample.txt'. Then, 'git status' is run, showing 'On branch master' and 'Untracked files: (use "git add <file>..." to include in what will be committed) sam.txt'. Then, 'git log' is run, showing 'nothing added to commit but untracked files present (use "git add" to track)'. Finally, 'git log' is run again, showing 'commit 3f244833711710597cf7bb2d16623a21550926fb (HEAD -> master)' and 'Author: abhi <abhi@ttn> Date: Wed Feb 3 12:09:30 2021 +0530'. The prompt 'abhi@ubuntu:~/project\$' is shown at the bottom.

```
abhi@ubuntu:~/project$ git config --global user.email "abhi@ttn"
abhi@ubuntu:~/project$ git config --global user.name "abhi"
abhi@ubuntu:~/project$ git commit -m "initial commit"
[master (root-commit) 3f24483] initial commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 sample.txt
abhi@ubuntu:~/project$ git status
On branch master

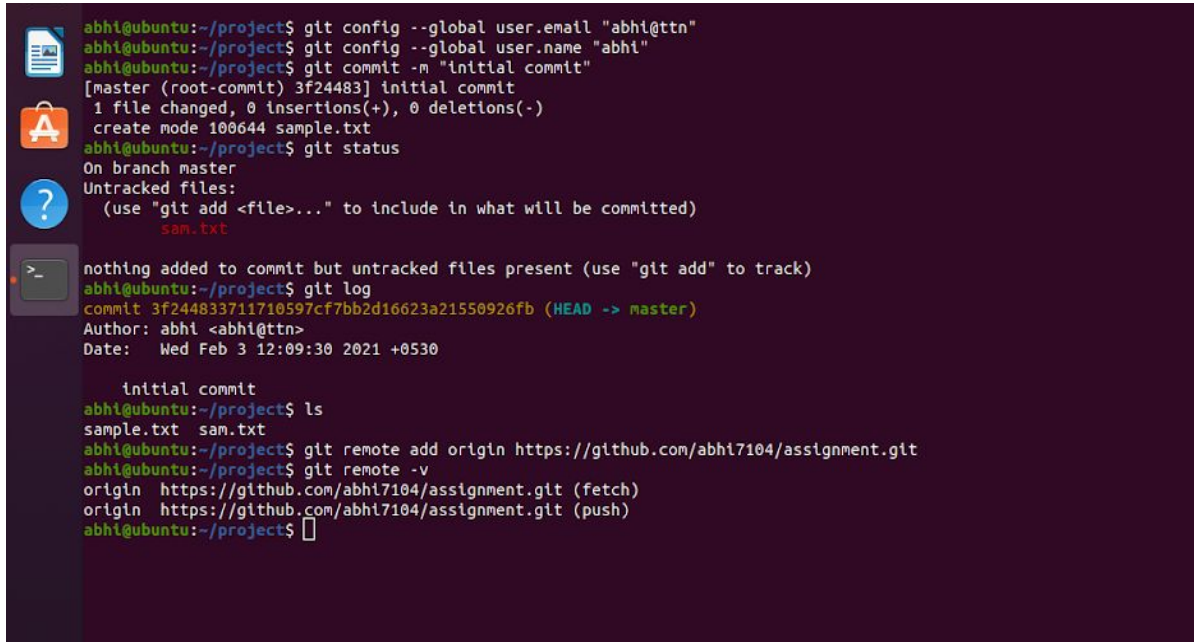
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        sam.txt

nothing added to commit but untracked files present (use "git add" to track)
abhi@ubuntu:~/project$ git log
commit 3f244833711710597cf7bb2d16623a21550926fb (HEAD -> master)
Author: abhi <abhi@ttn>
Date: Wed Feb 3 12:09:30 2021 +0530

    initial commit
abhi@ubuntu:~/project$
```

6. Add a remote

Command: `git commit -m "initial commit"`

A terminal window with a dark purple background and light green text. On the left side, there is a vertical sidebar with icons for a document, a folder, a question mark, and a terminal. The terminal output shows the following commands and their results:

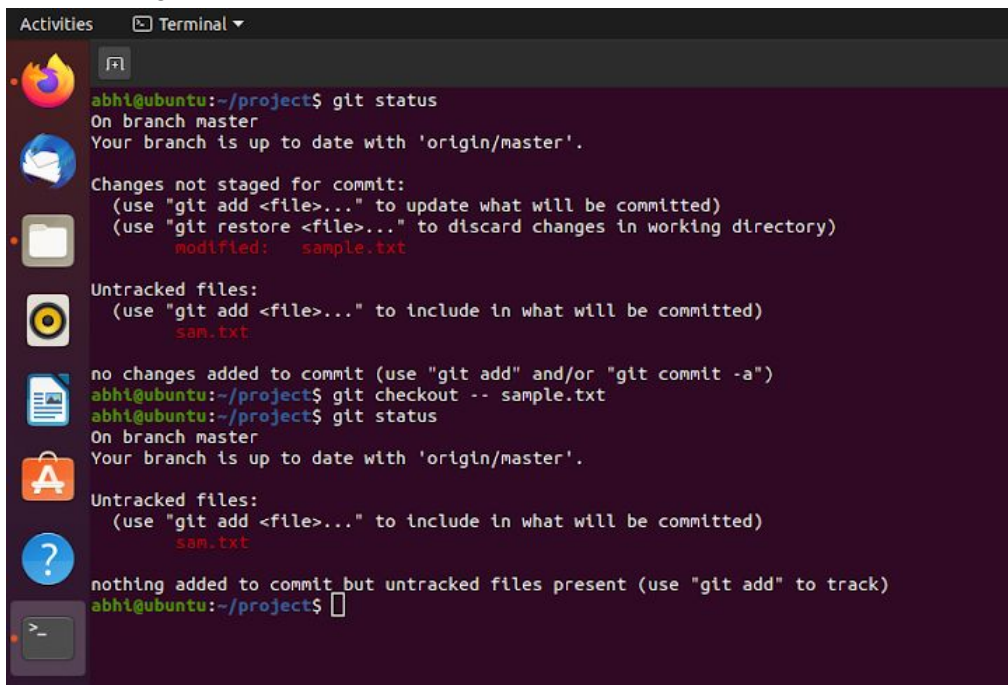
```
abhi@ubuntu:~/project$ git config --global user.email "abhi@ttn"
abhi@ubuntu:~/project$ git config --global user.name "abhi"
abhi@ubuntu:~/project$ git commit -m "initial commit"
[master (root-commit) 3f24483] initial commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 sample.txt
abhi@ubuntu:~/project$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
       sample.txt

nothing added to commit but untracked files present (use "git add" to track)
abhi@ubuntu:~/project$ git log
commit 3f244833711710597cf7bb2d16623a21550926fb (HEAD -> master)
Author: abhi <abhi@ttn>
Date:   Wed Feb 3 12:09:30 2021 +0530

    initial commit
abhi@ubuntu:~/project$ ls
sample.txt  sample.txt
abhi@ubuntu:~/project$ git remote add origin https://github.com/abhi7104/assignment.git
abhi@ubuntu:~/project$ git remote -v
origin  https://github.com/abhi7104/assignment.git (fetch)
origin  https://github.com/abhi7104/assignment.git (push)
abhi@ubuntu:~/project$
```

7. Undo changes to a particular file

Command: `git checkout -- sample.txt`

A terminal window with a dark purple background and light green text. On the left side, there is a vertical sidebar with icons for a document, a folder, a question mark, and a terminal. The terminal output shows the following commands and their results:

```
abhi@ubuntu:~/project$ 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:   sample.txt

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

no changes added to commit (use "git add" and/or "git commit -a")
abhi@ubuntu:~/project$ git checkout -- sample.txt
abhi@ubuntu:~/project$ git status
On branch master
Your branch is up to date with 'origin/master'.

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

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


8. Push changes to Github

Command: git push

```
abhi@ubuntu:~/project$ 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)
        modified:   sample.txt

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

abhi@ubuntu:~/project$ cat sample.txt
hello
thier
from
123456
abhi@ubuntu:~/project$ git commit -m "newlyupdated"
[master dfa0b63] newlyupdated
 1 file changed, 1 insertion(+)
abhi@ubuntu:~/project$ git push
Username for 'https://github.com': abhi7104
Password for 'https://abhi7104@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 254 bytes | 254.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/abhi7104/assignment.git
   b69a8b6..dfa0b63  master -> master
abhi@ubuntu:~/project$
```

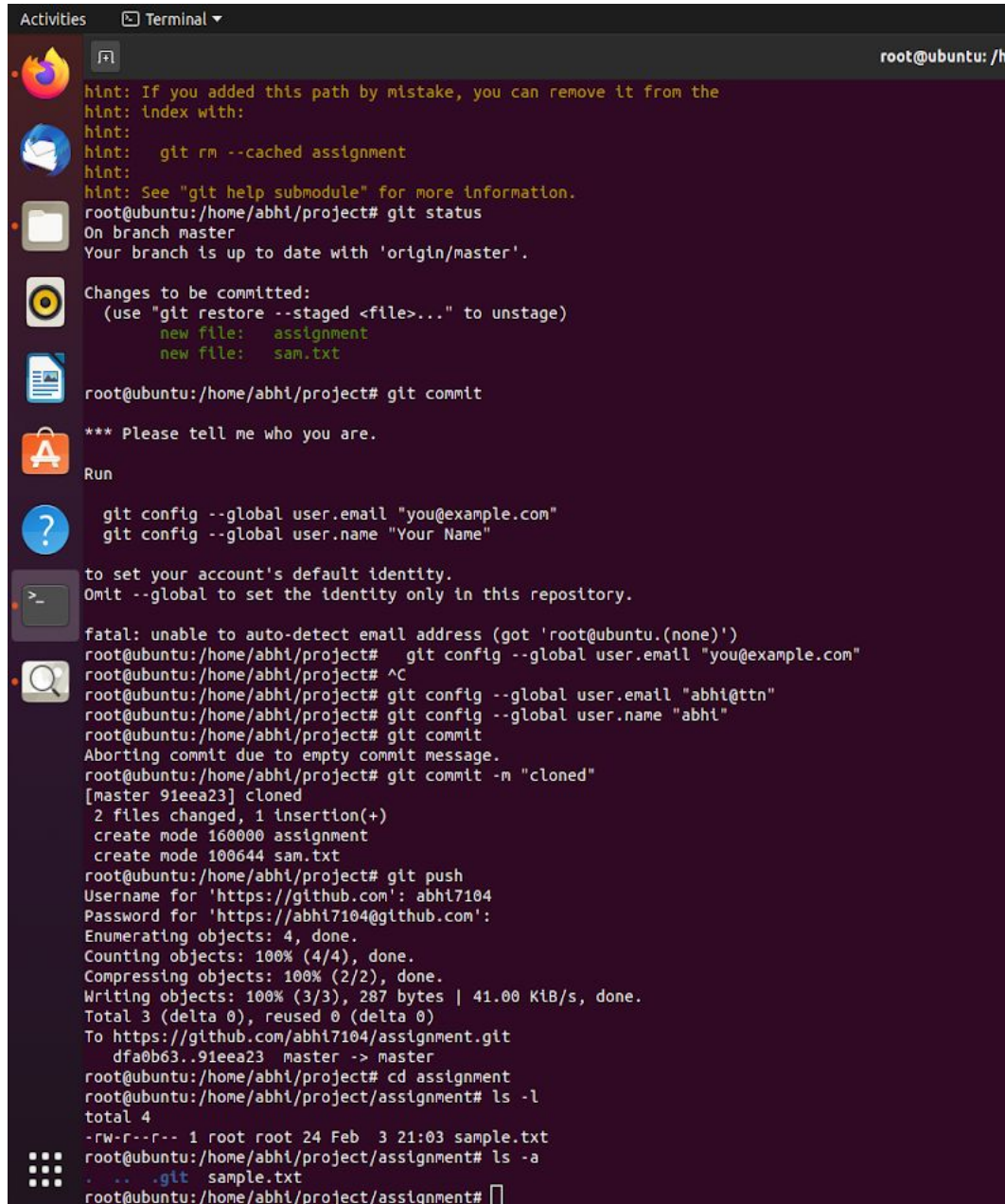
9. Clone the repository

Command: git clone <https://github.com/abhi7104/assignment.git>

```
Activities  Terminal
abhi@u
abhi@ubuntu:~$ git clone https://github.com/abhi7104/Module-2-Coding-Assignment.git
Cloning into 'Module-2-Coding-Assignment'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 5 (delta 0), pack-reused 0
Unpacking objects: 100% (8/8), 2.09 KiB | 535.00 KiB/s, done.
abhi@ubuntu:~$ ls
archive.tar.gz  Desktop  Documents  Downloads  exercise  Module-2-Coding-Assignment  Music
abhi@ubuntu:~$ cd Module-2-Coding-Assignment
abhi@ubuntu:~/Module-2-Coding-Assignment$ ls -l
total 8
drwxrwxr-x 2 abhi abhi 4096 Feb  3 16:52 css
-rw-rw-r-- 1 abhi abhi 1467 Feb  3 16:52 index.html
abhi@ubuntu:~/Module-2-Coding-Assignment$
```

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

Command: `git add .`
`git commit -m "cloned"`
`git push`

A terminal window titled 'Terminal' with a dark background and light text. The prompt is 'root@ubuntu: /h'. The output shows a series of git commands and their results. It starts with a hint about removing a path, followed by 'git status' showing the branch is up to date. Then 'git commit' is run, showing changes to 'assignment' and 'sam.txt'. This is followed by a prompt for user information, which is skipped. Then 'git config' is used to set a global email and name. An attempt to commit without a message fails. Then 'git commit -m "cloned"' is successful. Finally, 'git push' is run, showing the upload to a remote repository and a directory listing of the local 'assignment' folder.

```
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached assignment
hint:
hint: See "git help submodule" for more information.
root@ubuntu:/home/abhi/project# 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)
        new file:   assignment
        new file:   sam.txt

root@ubuntu:/home/abhi/project# git commit

*** Please tell me who you are.

Run

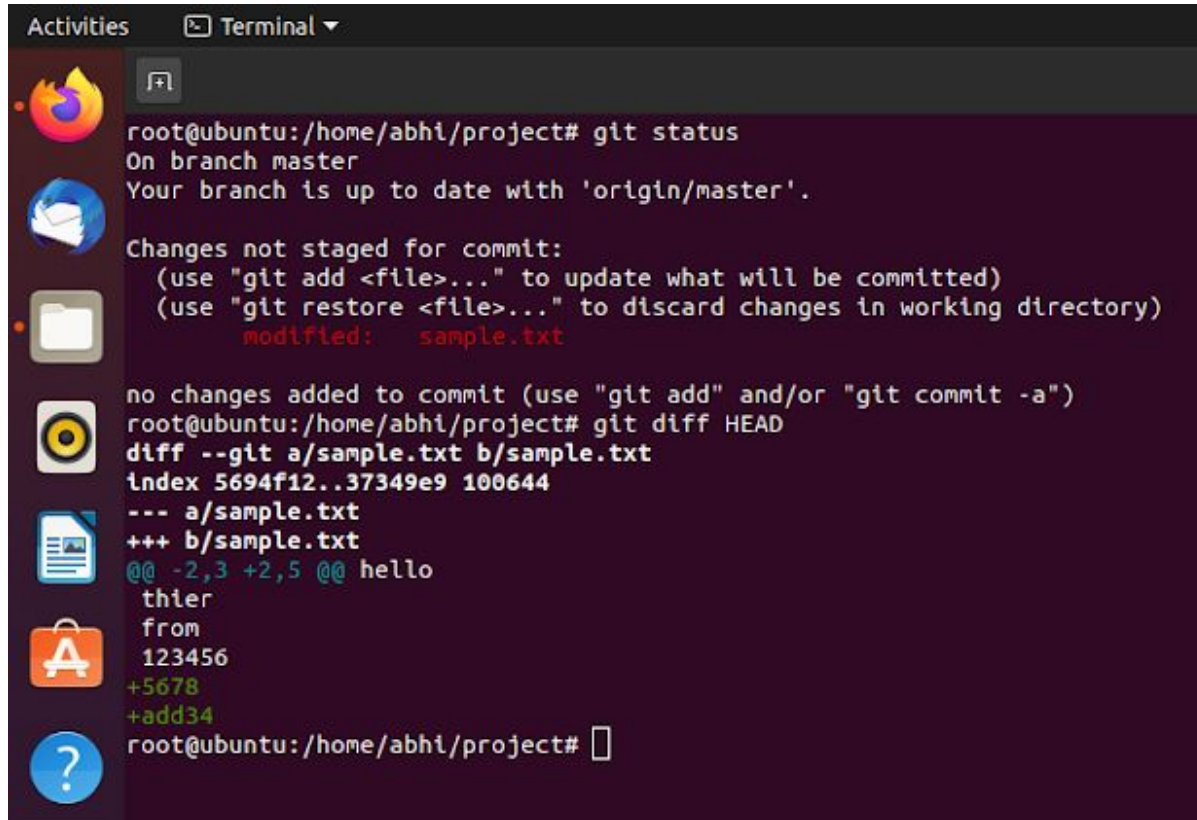
  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'root@ubuntu.(none)')
root@ubuntu:/home/abhi/project# git config --global user.email "you@example.com"
root@ubuntu:/home/abhi/project# ^C
root@ubuntu:/home/abhi/project# git config --global user.email "abhi@ttn"
root@ubuntu:/home/abhi/project# git config --global user.name "abhi"
root@ubuntu:/home/abhi/project# git commit
Aborting commit due to empty commit message.
root@ubuntu:/home/abhi/project# git commit -m "cloned"
[master 91eea23] cloned
 2 files changed, 1 insertion(+)
 create mode 160000 assignment
 create mode 100644 sam.txt
root@ubuntu:/home/abhi/project# git push
Username for 'https://github.com': abhi7104
Password for 'https://abhi7104@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 287 bytes | 41.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/abhi7104/assignment.git
   dfa0b63..91eea23  master -> master
root@ubuntu:/home/abhi/project# cd assignment
root@ubuntu:/home/abhi/project/assignment# ls -l
total 4
-rw-r--r-- 1 root root 24 Feb  3 21:03 sample.txt
root@ubuntu:/home/abhi/project/assignment# ls -a
.  ..  .git  sample.txt
root@ubuntu:/home/abhi/project/assignment#
```

11. Check differences between a file and its staged version

Command: git diff HEAD

A screenshot of a Linux terminal window. The terminal title bar shows 'Activities' and 'Terminal'. On the left side of the terminal, there is a vertical dock with icons for Firefox, a mail client, a file manager, a CD/DVD icon, a document icon, an application store icon, and a help icon. The terminal text shows the user running 'git status' and 'git diff HEAD' commands. The output of 'git status' indicates the branch is up to date with 'origin/master' and lists 'sample.txt' as a modified file not staged for commit. The output of 'git diff HEAD' shows the differences between the current working directory and the HEAD commit for 'a/sample.txt' and 'b/sample.txt'.

```
root@ubuntu:/home/abhi/project# 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:   sample.txt

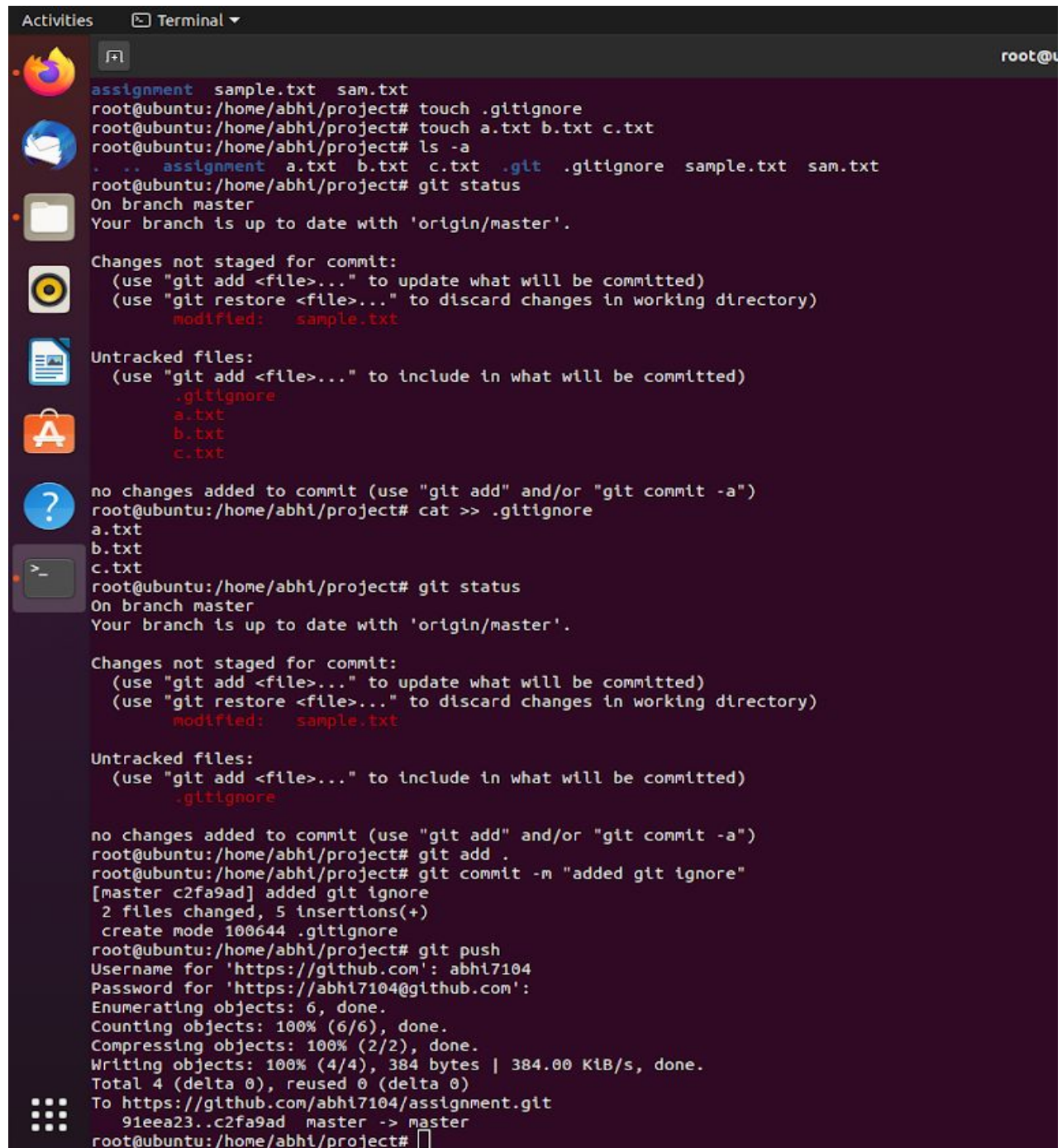
no changes added to commit (use "git add" and/or "git commit -a")
root@ubuntu:/home/abhi/project# git diff HEAD
diff --git a/sample.txt b/sample.txt
index 5694f12..37349e9 100644
--- a/sample.txt
+++ b/sample.txt
@@ -2,3 +2,5 @@ hello
  thier
  from
  123456
+5678
+add34
root@ubuntu:/home/abhi/project#
```

12. Ignore a few files to be checked in

Command: touch .gitignore

cat >> .gitignore

A.txt b.txt c.txt



```
Activities Terminal root@u
assignment sample.txt sam.txt
root@ubuntu:/home/abhi/project# touch .gitignore
root@ubuntu:/home/abhi/project# touch a.txt b.txt c.txt
root@ubuntu:/home/abhi/project# ls -la
. . . assignment a.txt b.txt c.txt .git .gitignore sample.txt sam.txt
root@ubuntu:/home/abhi/project# 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:   sample.txt

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

no changes added to commit (use "git add" and/or "git commit -a")
root@ubuntu:/home/abhi/project# cat >> .gitignore
a.txt
b.txt
c.txt
root@ubuntu:/home/abhi/project# git status
On branch master
Your branch is up to date with 'origin/master'.

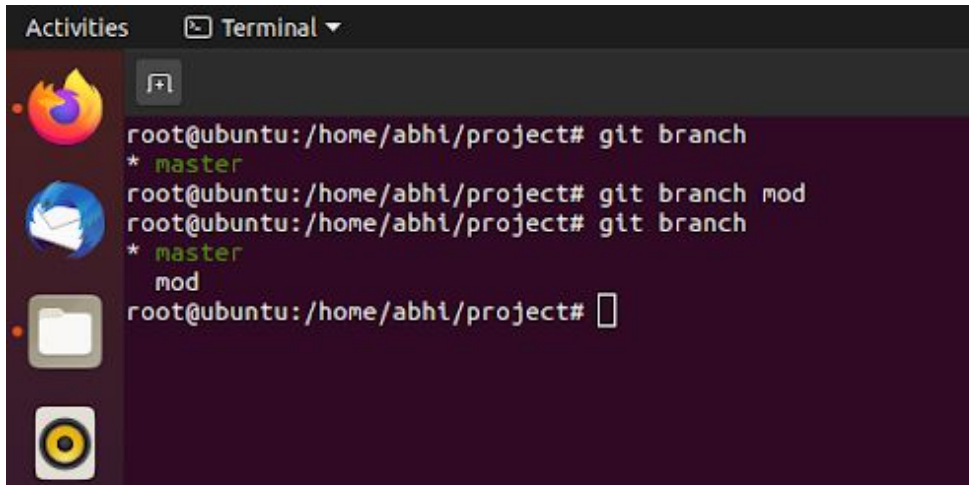
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   sample.txt

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")
root@ubuntu:/home/abhi/project# git add .
root@ubuntu:/home/abhi/project# git commit -m "added git ignore"
[master c2fa9ad] added git ignore
2 files changed, 5 insertions(+)
create mode 100644 .gitignore
root@ubuntu:/home/abhi/project# git push
Username for 'https://github.com': abhi7104
Password for 'https://abhi7104@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 384 bytes | 384.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/abhi7104/assignment.git
91eea23..c2fa9ad master -> master
root@ubuntu:/home/abhi/project#
```


13. Create a new branch.

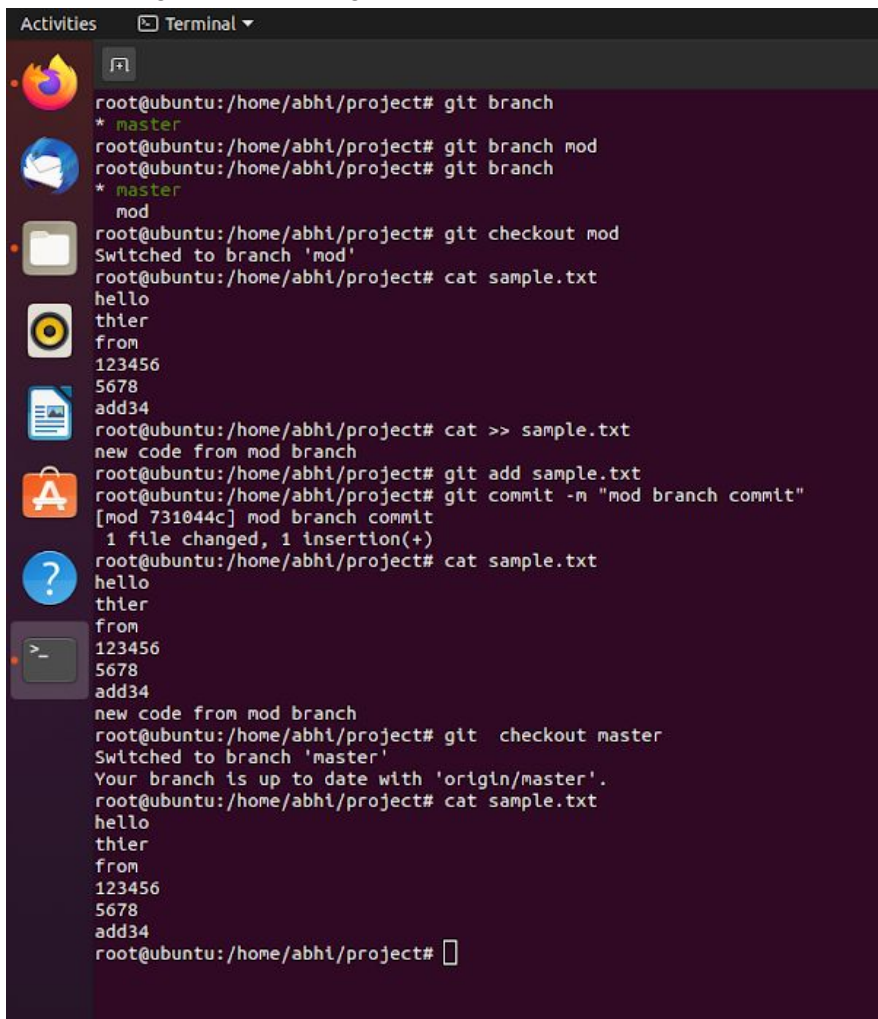
Command: git branch mod

A terminal window titled 'Terminal' showing the following commands and output:

```
root@ubuntu:/home/abhi/project# git branch
* master
root@ubuntu:/home/abhi/project# git branch mod
root@ubuntu:/home/abhi/project# git branch
* master
  mod
root@ubuntu:/home/abhi/project#
```

14. Diverge them with commits

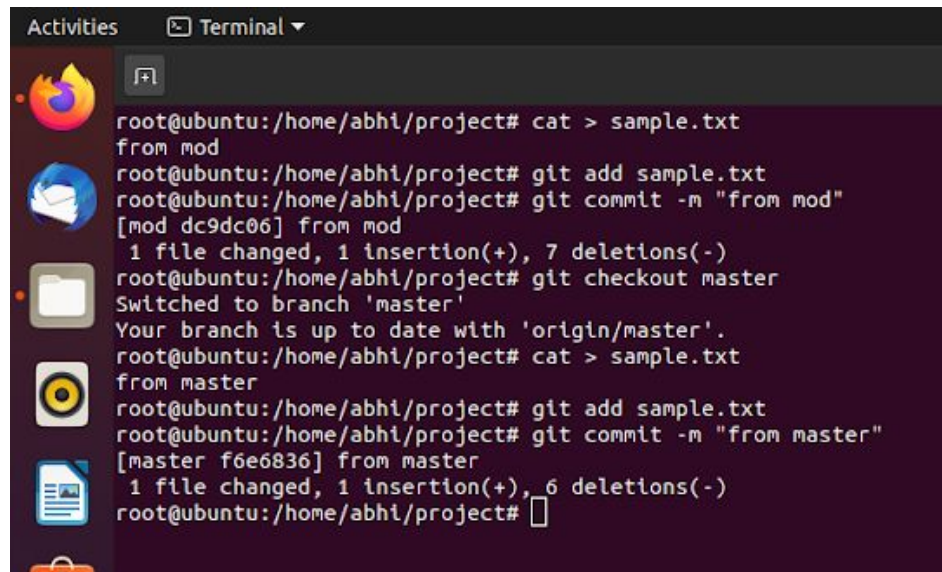
Command: git brach mod, git checkout mod

A terminal window titled 'Terminal' showing the following commands and output:

```
root@ubuntu:/home/abhi/project# git branch
* master
root@ubuntu:/home/abhi/project# git branch mod
root@ubuntu:/home/abhi/project# git branch
* master
  mod
root@ubuntu:/home/abhi/project# git checkout mod
Switched to branch 'mod'
root@ubuntu:/home/abhi/project# cat sample.txt
hello
thier
from
123456
5678
add34
root@ubuntu:/home/abhi/project# cat >> sample.txt
new code from mod branch
root@ubuntu:/home/abhi/project# git add sample.txt
root@ubuntu:/home/abhi/project# git commit -m "mod branch commit"
[mod 731044c] mod branch commit
1 file changed, 1 insertion(+)
root@ubuntu:/home/abhi/project# cat sample.txt
hello
thier
from
123456
5678
add34
new code from mod branch
root@ubuntu:/home/abhi/project# git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
root@ubuntu:/home/abhi/project# cat sample.txt
hello
thier
from
123456
5678
add34
root@ubuntu:/home/abhi/project#
```

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

Command: git checkout mod , git checkout master
cat > sample.txt , cat > sample.txt
git add sample.txt , git add sample.txt
git commit -m "from mod" ,git commit -m "from master"

A screenshot of a Linux terminal window titled 'Terminal'. The terminal shows a sequence of git commands and their outputs. The user starts in the 'mod' branch, creates a file 'sample.txt', adds it to git, and commits it with the message 'from mod'. Then, they checkout to the 'master' branch, which is up to date with 'origin/master'. They edit 'sample.txt' again, add it to git, and commit it with the message 'from master'. The terminal output shows the commit hashes and file changes for both commits.

```
root@ubuntu:/home/abhi/project# cat > sample.txt
from mod
root@ubuntu:/home/abhi/project# git add sample.txt
root@ubuntu:/home/abhi/project# git commit -m "from mod"
[mod dc9dc06] from mod
1 file changed, 1 insertion(+), 7 deletions(-)
root@ubuntu:/home/abhi/project# git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
root@ubuntu:/home/abhi/project# cat > sample.txt
from master
root@ubuntu:/home/abhi/project# git add sample.txt
root@ubuntu:/home/abhi/project# git commit -m "from master"
[master f6e6836] from master
1 file changed, 1 insertion(+), 6 deletions(-)
root@ubuntu:/home/abhi/project#
```

16. Try merging and resolve merge conflicts

Command: git merge mod

```
es  Terminal ▾

root@ubuntu:/home/abhi/project# git merge mod
Auto-merging sample.txt
CONFLICT (content): Merge conflict in sample.txt
Automatic merge failed; fix conflicts and then commit the result.
root@ubuntu:/home/abhi/project# git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   sample.txt

no changes added to commit (use "git add" and/or "git commit -a")
root@ubuntu:/home/abhi/project# cat sample.txt
<<<<<< HEAD
from master
=====
from mod
>>>>>> mod
root@ubuntu:/home/abhi/project# nano sample.txt
root@ubuntu:/home/abhi/project# git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

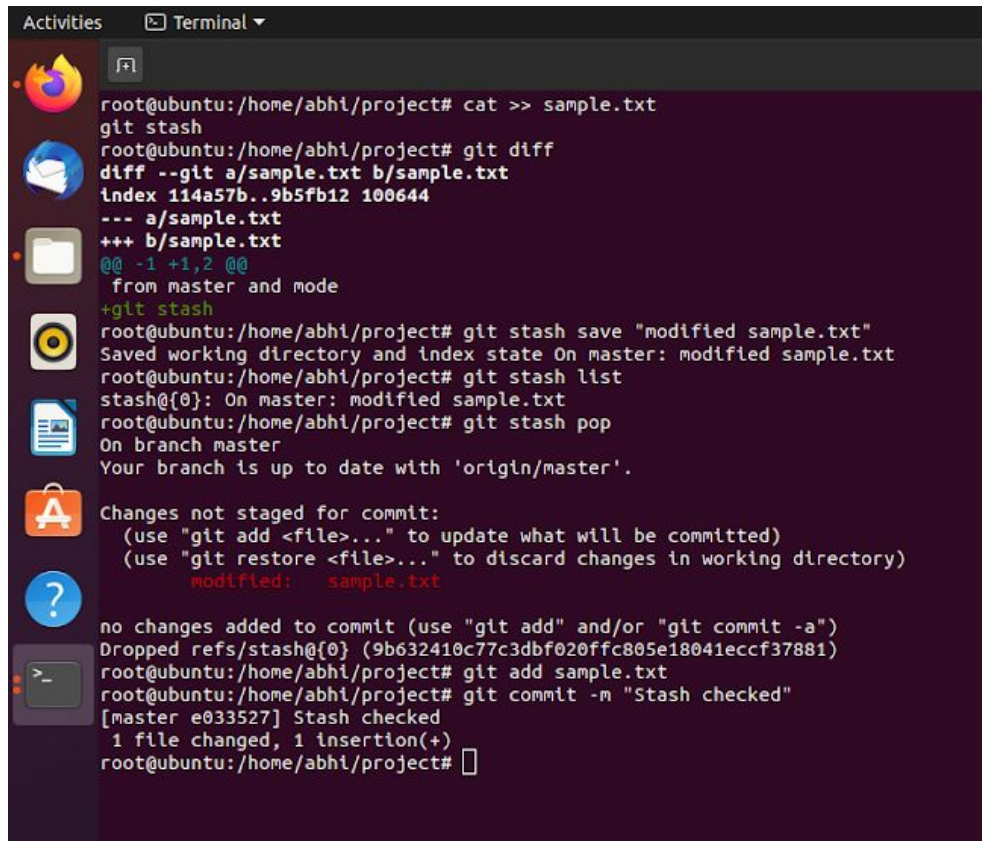
Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   sample.txt

no changes added to commit (use "git add" and/or "git commit -a")
root@ubuntu:/home/abhi/project# cat sample.txt
from master and mode
root@ubuntu:/home/abhi/project# git add sample.txt
root@ubuntu:/home/abhi/project# git commit -m "udated branch"
[master 8d89e9d] udated branch
root@ubuntu:/home/abhi/project# git push

root@ubuntu:/home/abhi/project# git push
Username for 'https://github.com': abhi7104
Password for 'https://abhi7104@github.com':
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Compressing objects: 100% (9/9), done.
Writing objects: 100% (12/12), 976 bytes | 162.00 KiB/s, done.
Total 12 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/abhi7104/assignment.git
   c2fa9ad..8d89e9d  master -> master
root@ubuntu:/home/abhi/project#
```

17. Stash the changes and pop them

Command: `git stash save "modified sample.txt"`
`git stash list`
`git stash pop`

A terminal window titled 'Terminal' showing a series of git commands and their outputs. The user is in a directory /home/abhi/project. The workflow includes: creating a file sample.txt, stashing the changes with a message, listing the stash, popping the stash, and finally committing the changes. The terminal output shows the diff between the master branch and the stash, the stash being saved, listed, and then popped, and the final commit message 'Stash checked'.

```
root@ubuntu:/home/abhi/project# cat >> sample.txt
git stash
root@ubuntu:/home/abhi/project# git diff
diff --git a/sample.txt b/sample.txt
index 114a57b..9b5fb12 100644
--- a/sample.txt
+++ b/sample.txt
@@ -1,2 @@
 from master and mode
+git stash
root@ubuntu:/home/abhi/project# git stash save "modified sample.txt"
Saved working directory and index state On master: modified sample.txt
root@ubuntu:/home/abhi/project# git stash list
stash@{0}: On master: modified sample.txt
root@ubuntu:/home/abhi/project# git stash pop
On branch master
Your branch is up to date with 'origin/master'.

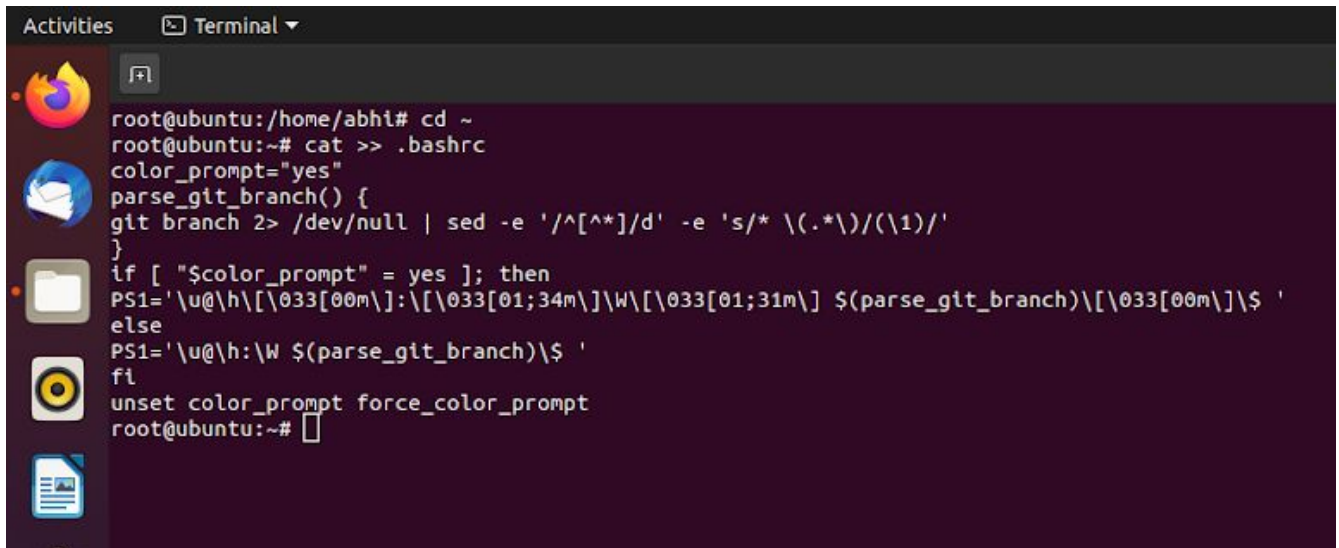
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   sample.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (9b632410c77c3dbf020ffc805e18041eccf37881)
root@ubuntu:/home/abhi/project# git add sample.txt
root@ubuntu:/home/abhi/project# git commit -m "Stash checked"
[master e033527] Stash checked
1 file changed, 1 insertion(+)
root@ubuntu:/home/abhi/project#
```


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
```

Command: cat >> .bashrc



The screenshot shows a terminal window with the following commands and output:

```
root@ubuntu:/home/abhi# cd ~  
root@ubuntu:~# cat >> .bashrc  
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  
root@ubuntu:~#
```