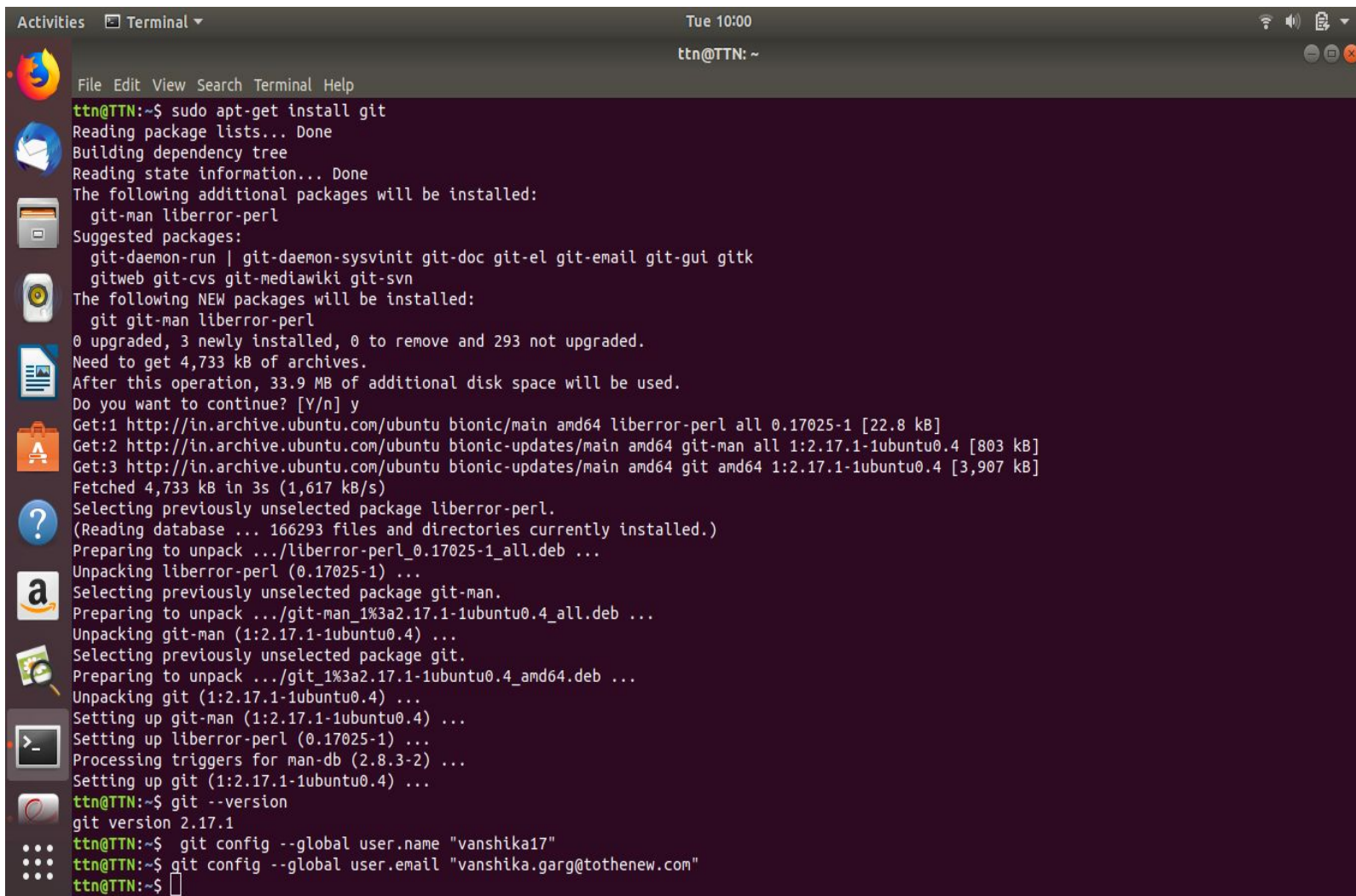


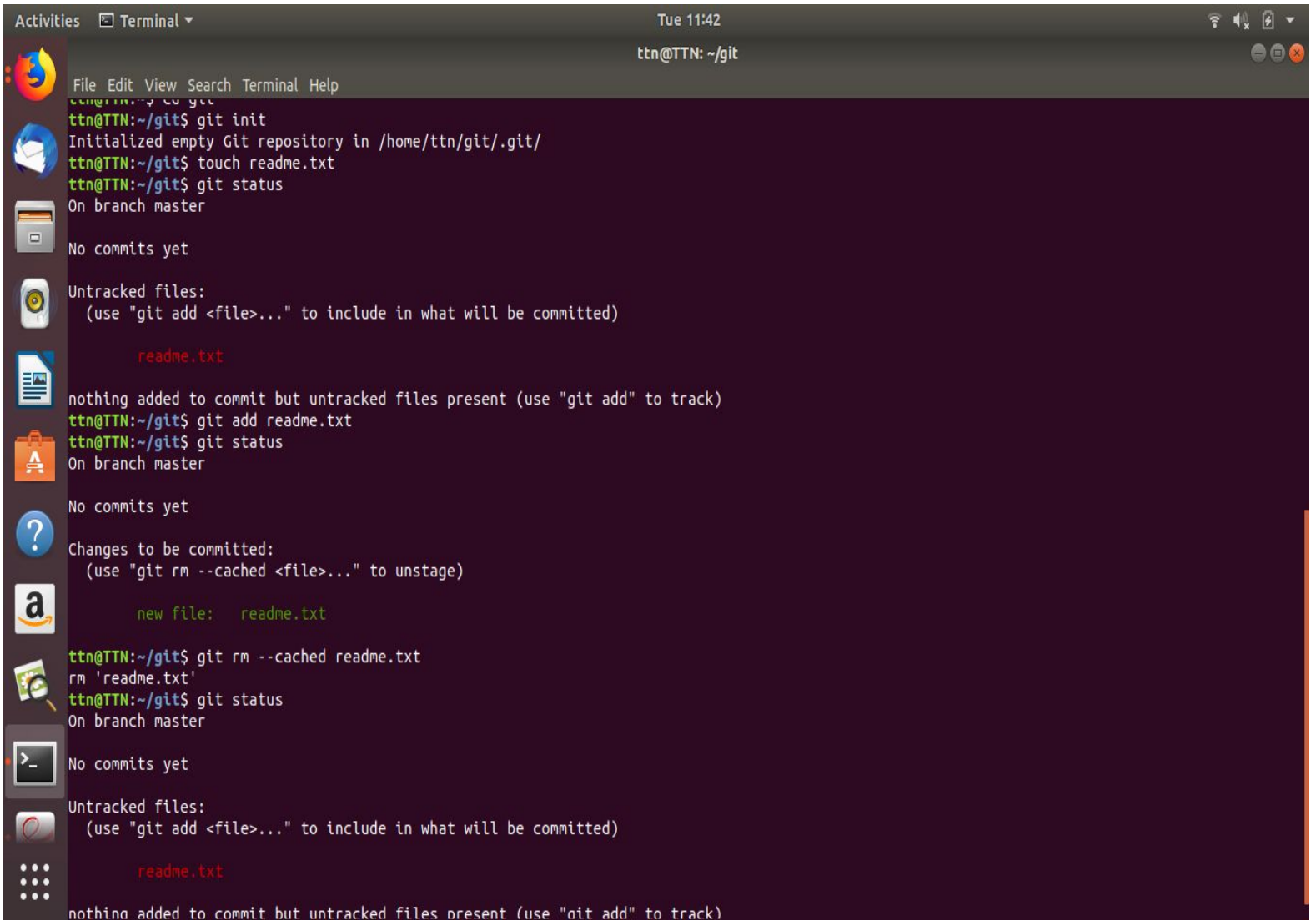
Assessment - 2

1. Git Setup



```
ttn@TTN:~$ sudo apt-get install git
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 293 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 (1,617 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 166293 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17025-1_all.deb ...
Unpacking liberror-perl (0.17025-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.17.1-1ubuntu0.4_all.deb ...
Unpacking git-man (1:2.17.1-1ubuntu0.4) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.17.1-1ubuntu0.4_amd64.deb ...
Unpacking git (1:2.17.1-1ubuntu0.4) ...
Setting up git-man (1:2.17.1-1ubuntu0.4) ...
Setting up liberror-perl (0.17025-1) ...
Processing triggers for man-db (2.8.3-2) ...
Setting up git (1:2.17.1-1ubuntu0.4) ...
ttn@TTN:~$ git --version
git version 2.17.1
ttn@TTN:~$ git config --global user.name "vanshika17"
ttn@TTN:~$ git config --global user.email "vanshika.garg@tothenew.com"
ttn@TTN:~$
```

2. Initialize a Git Repository
3. Add files to the repository
4. Unstage 1 file



The terminal window shows the following sequence of commands and outputs:

```
File Edit View Search Terminal Help
ttn@TTN: ~/git
ttn@TTN:~/git$ git init
Initialized empty Git repository in /home/ttn/git/.git/
ttn@TTN:~/git$ touch readme.txt
ttn@TTN:~/git$ git status
On branch master

No commits yet

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

        readme.txt

nothing added to commit but untracked files present (use "git add" to track)
ttn@TTN:~/git$ git add readme.txt
ttn@TTN:~/git$ git status
On branch master

No commits yet

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

        new file:   readme.txt

ttn@TTN:~/git$ git rm --cached readme.txt
rm 'readme.txt'
ttn@TTN:~/git$ git status
On branch master

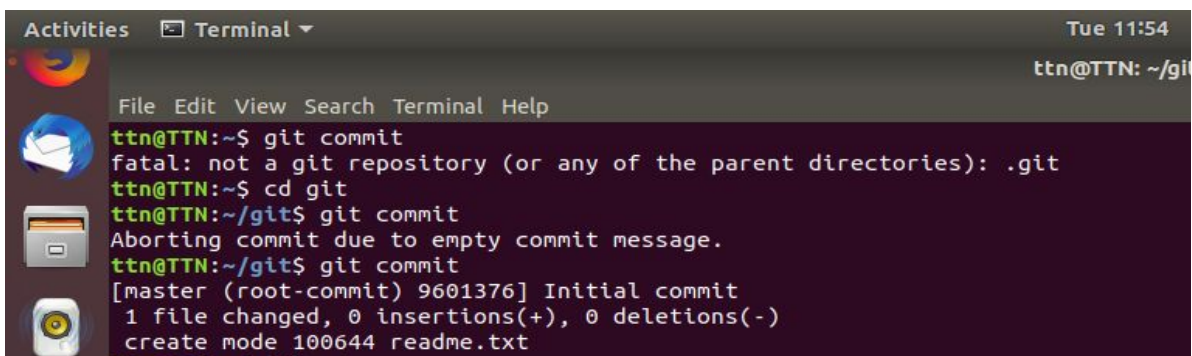
No commits yet

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

        readme.txt

nothing added to commit but untracked files present (use "git add" to track)
```

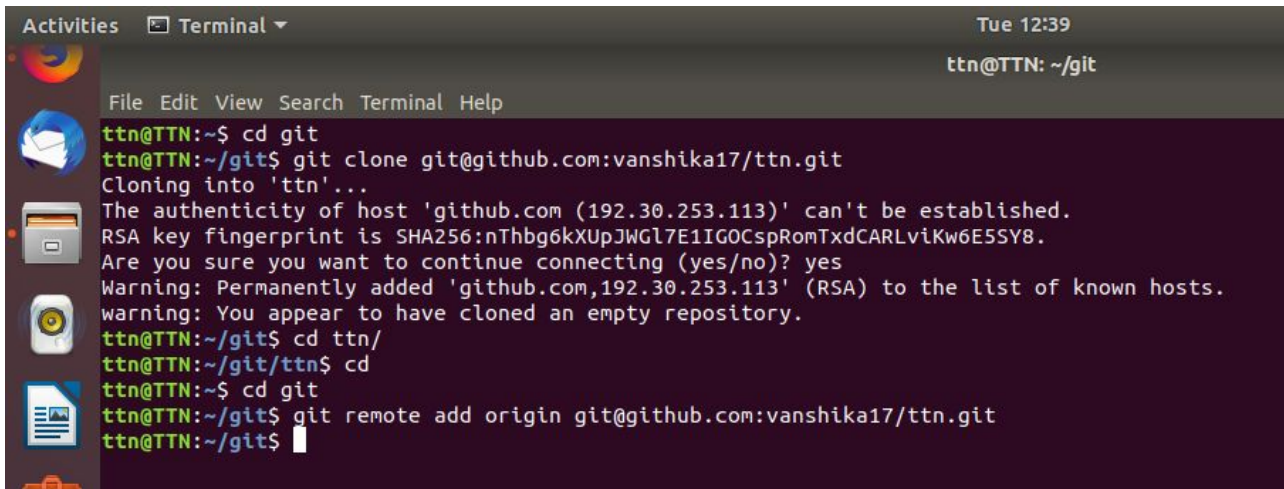
5. Commit a file



The terminal window shows the following sequence of commands and outputs:

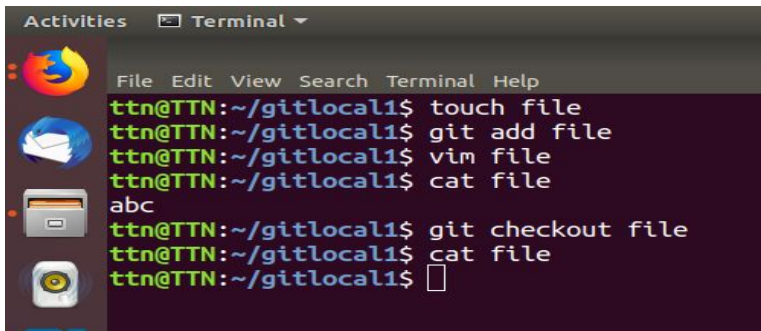
```
File Edit View Search Terminal Help
ttn@TTN:~$ git commit
fatal: not a git repository (or any of the parent directories): .git
ttn@TTN:~$ cd git
ttn@TTN:~/git$ git commit
Aborting commit due to empty commit message.
ttn@TTN:~/git$ git commit
[master (root-commit) 9601376] Initial commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 readme.txt
ttn@TTN:~/git$
```

6. Add a remote



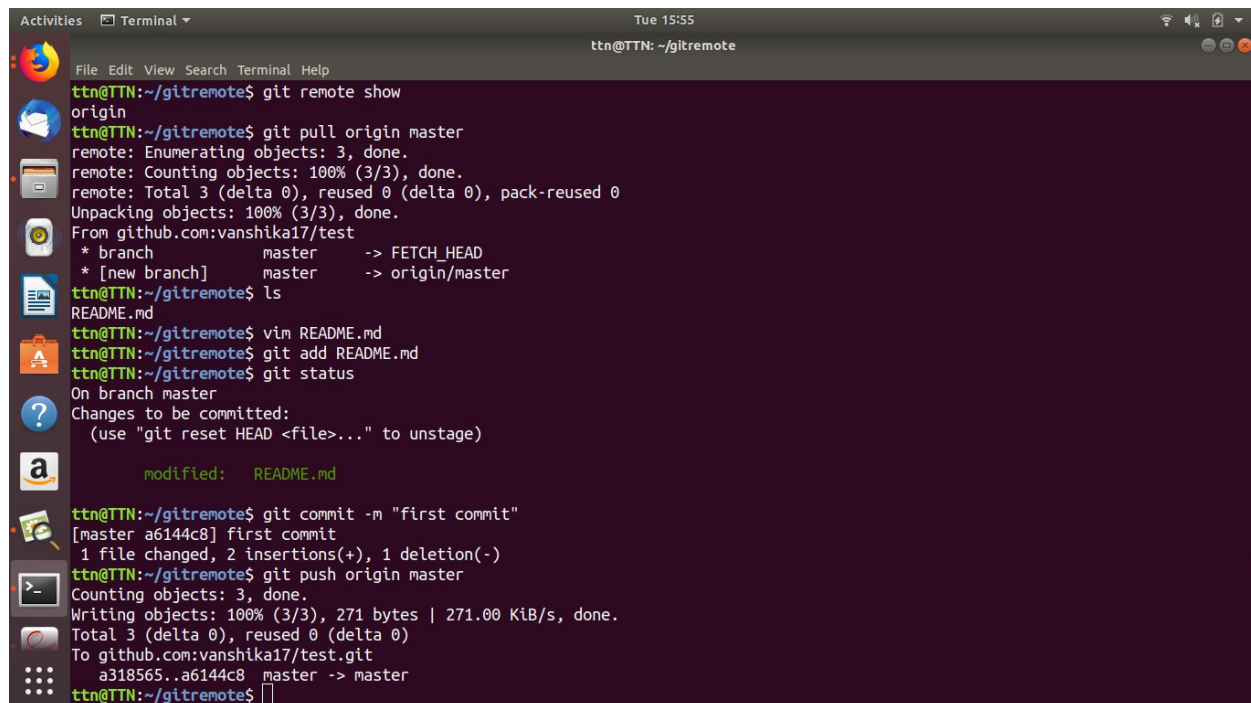
```
Activities Terminal Tue 12:39
ttn@TTN: ~/git
File Edit View Search Terminal Help
ttn@TTN:~$ cd git
ttn@TTN:~/git$ git clone git@github.com:vanshika17/ttn.git
Cloning into 'ttn'...
The authenticity of host 'github.com (192.30.253.113)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGL7E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.253.113' (RSA) to the list of known hosts.
warning: You appear to have cloned an empty repository.
ttn@TTN:~/git$ cd ttn/
ttn@TTN:~/git/ttn$ cd
ttn@TTN:~$ cd git
ttn@TTN:~/git$ git remote add origin git@github.com:vanshika17/ttn.git
ttn@TTN:~/git$
```

7. Undo changes to a file



```
Activities Terminal
ttn@TTN:~/gitlocal1$ touch file
ttn@TTN:~/gitlocal1$ git add file
ttn@TTN:~/gitlocal1$ vim file
ttn@TTN:~/gitlocal1$ cat file
abc
ttn@TTN:~/gitlocal1$ git checkout file
ttn@TTN:~/gitlocal1$ cat file
ttn@TTN:~/gitlocal1$
```

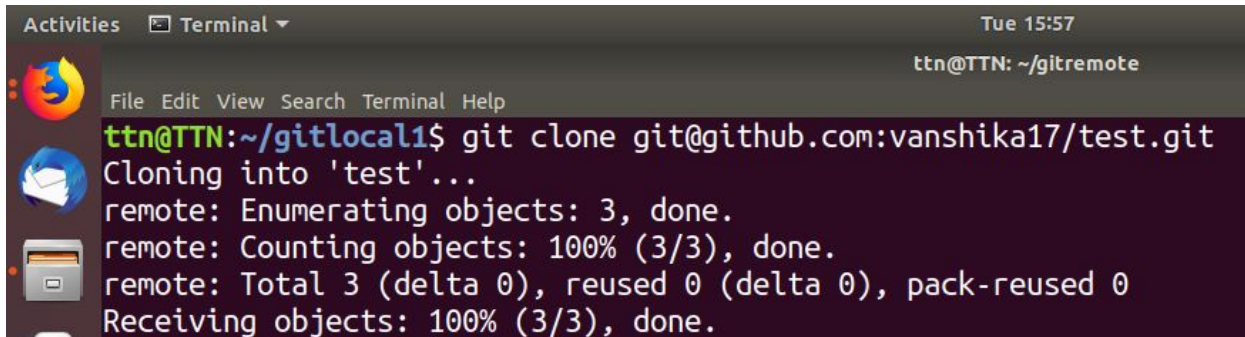
8. Push changes to Github



```
Activities Terminal Tue 15:55
ttn@TTN: ~/gitremote
File Edit View Search Terminal Help
ttn@TTN:~/gitremote$ git remote show
origin
ttn@TTN:~/gitremote$ git pull origin master
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:vanshika17/test
* branch          master       -> FETCH_HEAD
* [new branch]     master       -> origin/master
ttn@TTN:~/gitremote$ ls
README.md
ttn@TTN:~/gitremote$ vim README.md
ttn@TTN:~/gitremote$ git add README.md
ttn@TTN:~/gitremote$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

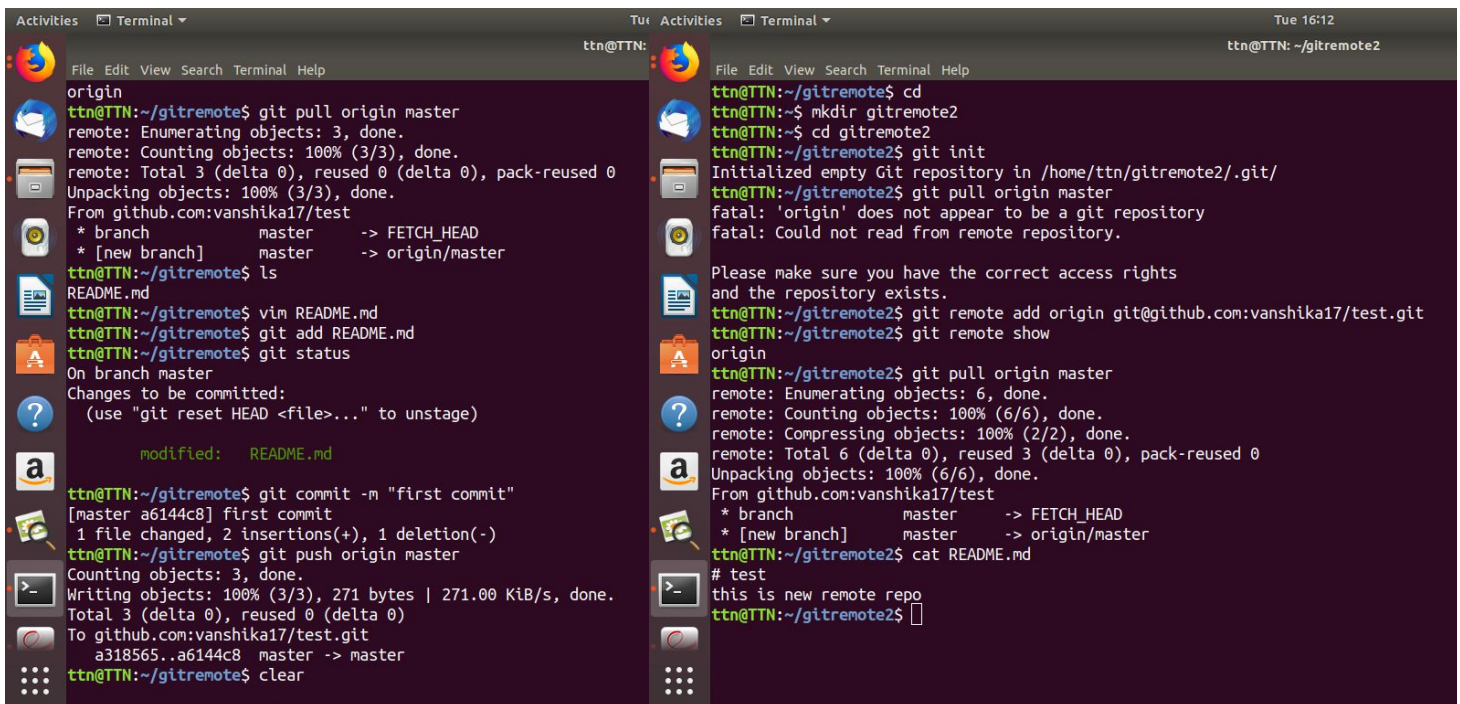
        modified:   README.md
ttn@TTN:~/gitremote$ git commit -m "first commit"
[master a6144c8] first commit
1 file changed, 2 insertions(+), 1 deletion(-)
ttn@TTN:~/gitremote$ git push origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 271 bytes | 271.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:vanshika17/test.git
   a318565..a6144c8  master -> master
ttn@TTN:~/gitremote$
```


9. Clone the repository

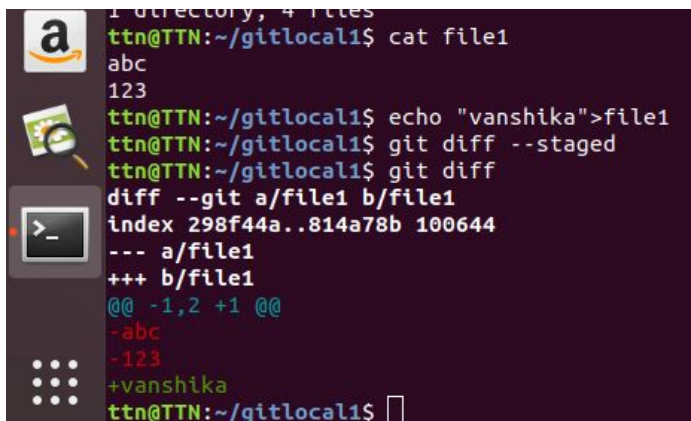
A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Tue 15:57, ttn@TTN: ~/gitremote). The prompt is ttn@TTN:~/gitlocal1\$. The command 'git clone git@github.com:vanshika17/test.git' is entered. The output shows the cloning process: 'Cloning into 'test'...', 'remote: Enumerating objects: 3, done.', 'remote: Counting objects: 100% (3/3), done.', 'remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0', and 'Receiving objects: 100% (3/3), done.'

```
File Edit View Search Terminal Help
ttn@TTN:~/gitlocal1$ git clone git@github.com:vanshika17/test.git
Cloning into 'test'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

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

Two terminal windows are shown side-by-side. The left window (ttn@TTN:~/gitremote) shows the process of pulling changes from the origin master branch. The right window (ttn@TTN:~/gitremote2) shows the process of adding a new file (README.md) and pushing it to the origin master branch. The left window's output includes: 'git pull origin master', 'remote: Enumerating objects: 3, done.', 'remote: Counting objects: 100% (3/3), done.', 'remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0', 'Unpacking objects: 100% (3/3), done.', 'From github.com:vanshika17/test', '* branch master -> FETCH_HEAD', '* [new branch] master -> origin/master', 'ttn@TTN:~/gitremote\$ ls', 'README.md', 'ttn@TTN:~/gitremote\$ vim README.md', 'ttn@TTN:~/gitremote\$ git add README.md', 'ttn@TTN:~/gitremote\$ git status', 'On branch master', 'Changes to be committed:', '(use "git reset HEAD <file>..." to unstage)', 'modified: README.md', 'ttn@TTN:~/gitremote\$ git commit -m "first commit"', '[master a6144c8] first commit', '1 file changed, 2 insertions(+), 1 deletion(-)', 'ttn@TTN:~/gitremote\$ git push origin master', 'Counting objects: 3, done.', 'Writing objects: 100% (3/3), 271 bytes | 271.00 KiB/s, done.', 'Total 3 (delta 0), reused 0 (delta 0)', 'To github.com:vanshika17/test.git', 'a318565..a6144c8 master -> master', 'ttn@TTN:~/gitremote\$ clear'. The right window's output includes: 'ttn@TTN:~/gitremote2\$ cd', 'ttn@TTN:~/gitremote2\$ mkdir gitremote2', 'ttn@TTN:~/gitremote2\$ cd gitremote2', 'ttn@TTN:~/gitremote2\$ git init', 'Initialized empty Git repository in /home/ttn/gitremote2/.git/', 'ttn@TTN:~/gitremote2\$ git pull origin master', 'fatal: 'origin' does not appear to be a git repository', 'fatal: Could not read from remote repository.', 'Please make sure you have the correct access rights and the repository exists.', 'ttn@TTN:~/gitremote2\$ git remote add origin git@github.com:vanshika17/test.git', 'ttn@TTN:~/gitremote2\$ git remote show', 'origin', 'ttn@TTN:~/gitremote2\$ git pull origin master', 'remote: Enumerating objects: 6, done.', 'remote: Counting objects: 100% (6/6), done.', 'remote: Compressing objects: 100% (2/2), done.', 'remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0', 'Unpacking objects: 100% (6/6), done.', 'From github.com:vanshika17/test', '* branch master -> FETCH_HEAD', '* [new branch] master -> origin/master', 'ttn@TTN:~/gitremote2\$ cat README.md', '# test', 'this is new remote repo', 'ttn@TTN:~/gitremote2\$'.

11. Check differences between a file and its staged version

A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Tue 15:57, ttn@TTN: ~/gitremote). The prompt is ttn@TTN:~/gitlocal1\$. The command 'git diff --staged' is entered. The output shows the differences between the staged version of 'file1' and the working directory version of 'file1'. The output is: 'diff --git a/file1 b/file1', 'index 298f44a..814a78b 100644', '--- a/file1', '+++ b/file1', '@@ -1,2 +1 @@', '-abc', '-123', '+vanshika', 'ttn@TTN:~/gitlocal1\$'.

12. Ignore a few files to be checked in

```
ttn@TTN:~/gitremote2$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   .gitignore

ttn@TTN:~/gitremote2$ touch file1 file2 file3
ttn@TTN:~/gitremote2$ vim .gitignore
ttn@TTN:~/gitremote2$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   .gitignore

ttn@TTN:~/gitremote2$
```

13. Create a new branch

```
Activities  Terminal ▾

File Edit View Search Terminal Help
ttn@TTN:~/gitlocal1$ git branch newbranch
ttn@TTN:~/gitlocal1$ git branch
* master
  newbranch
ttn@TTN:~/gitlocal1$
```

14. Diverge them with commits

```
Activities  Terminal ▾

File Edit View Search Terminal Help
ttn@TTN:~/repo$ git branch branch1
ttn@TTN:~/repo$ git branch
* master
  branch1
ttn@TTN:~/repo$ touch commitfile
ttn@TTN:~/repo$ echo "master branch edit">commitfile
ttn@TTN:~/repo$ git add commitfile
ttn@TTN:~/repo$ git commit -m "master branch commit"
[master 4c250e5] master branch commit
1 file changed, 1 insertion(+)
create mode 100644 commitfile
ttn@TTN:~/repo$ checkout branch1
checkout: command not found
ttn@TTN:~/repo$ git checkout branch1
Switched to branch 'branch1'
ttn@TTN:~/repo$ vim commitfile
ttn@TTN:~/repo$ clear

ttn@TTN:~/repo$ git checkout master
Switched to branch 'master'
ttn@TTN:~/repo$ vim commitfile
ttn@TTN:~/repo$ git checkout branch1
Switched to branch 'branch1'
ttn@TTN:~/repo$ vim commitfile
ttn@TTN:~/repo$ git add commitfile
ttn@TTN:~/repo$ git commit -m "branch1 commit"
[branch1 0b208cc] branch1 commit
1 file changed, 1 insertion(+)
create mode 100644 commitfile
ttn@TTN:~/repo$
```


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

```
Activities Terminal Tue 2
ttn@TTN:~/repo$ cat commitfile
branch file edit
ttn@TTN:~/repo$ git brnach
git: 'brnach' is not a git command. See 'git --help'.

The most similar command is
branch
ttn@TTN:~/repo$ git branch
* branch1
master
ttn@TTN:~/repo$ git checkout master
Switched to branch 'master'
ttn@TTN:~/repo$ cat commitfile
master branch edit
ttn@TTN:~/repo$ git branch
branch1
* master
ttn@TTN:~/repo$ git push origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 301 bytes | 301.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:vanshika17/testrepo.git
4578214..4c250e5 master -> master
ttn@TTN:~/repo$ git push origin branch1
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 296 bytes | 296.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'branch1' on GitHub by visiting:
remote: https://github.com/vanshika17/testrepo/pull/new/branch1
remote:
To github.com:vanshika17/testrepo.git
* [new branch] branch1 -> branch1
ttn@TTN:~/repo$
```

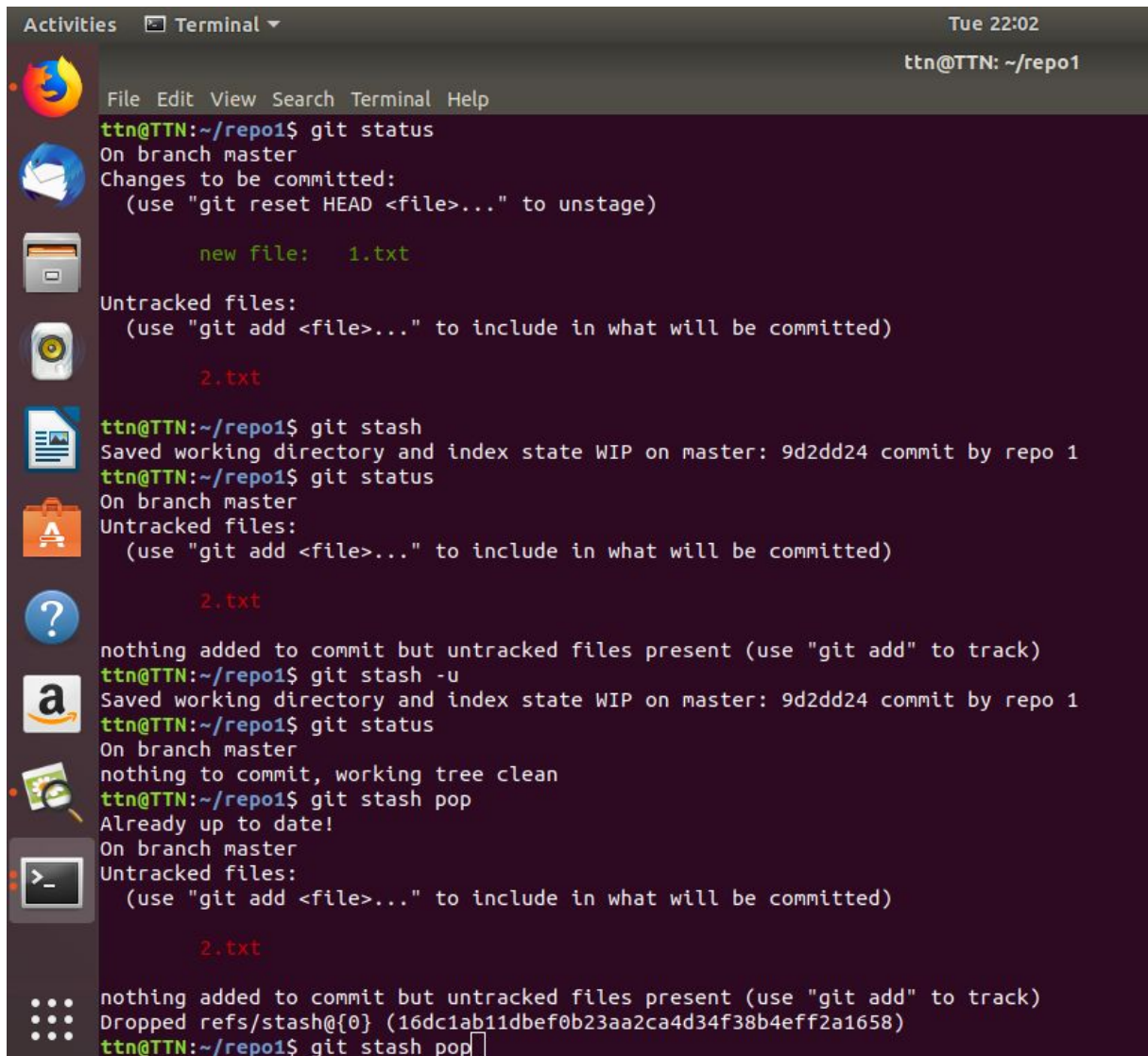
16. Try merging and resolve merge conflicts

```
Activities Terminal Tue 21:40
ttn@TTN: ~/repo1
File Edit View Search Terminal Help
ttn@TTN:~/repo1$ git remote add repo1 git@github.com:vanshika17/testrepo.git
ttn@TTN:~/repo1$ git pull repo1 master
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
From github.com:vanshika17/testrepo
* branch master -> FETCH_HEAD
* [new branch] master -> repo1/master
ttn@TTN:~/repo1$ ls
commitfile README.md
ttn@TTN:~/repo1$ vim README.md
ttn@TTN:~/repo1$ git add README.md
ttn@TTN:~/repo1$ git commit -m "commit by repo 1"
[master 9d2dd24] commit by repo 1
1 file changed, 2 insertions(+), 1 deletion(-)
ttn@TTN:~/repo1$ git push repo1 master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 312 bytes | 312.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:vanshika17/testrepo.git
4c250e5..9d2dd24 master -> master
ttn@TTN:~/repo1$

ttn@TTN: ~/repo2
File Edit View Search Terminal Help
ttn@TTN:~/repo2$ git remote add repo2 git@github.com:vanshika17/testrepo.git
ttn@TTN:~/repo2$ git pull repo2 master
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
From github.com:vanshika17/testrepo
* branch master -> FETCH_HEAD
* [new branch] master -> repo2/master
ttn@TTN:~/repo2$ ls
commitfile README.md
ttn@TTN:~/repo2$ vim README.md
ttn@TTN:~/repo2$ git add README.md
git: 'add' is not a git command. See 'git --help'.

The most similar command is
add
ttn@TTN:~/repo2$ git add README.md
ttn@TTN:~/repo2$ git commit -m "commit by repo2"
[master 96df4ff] commit by repo2
1 file changed, 2 insertions(+), 1 deletion(-)
ttn@TTN:~/repo2$ git push repo1 master
To github.com:vanshika17/testrepo.git
! [rejected] master -> master (fetch first)
error: failed to push some refs to 'git@github.com:vanshika17/testrepo.git'
hint: Updates were rejected because the remote contains work that you
do
hint: not have locally. This is usually caused by another repository p
ushing
hint: to the same ref. You may want to first integrate the remote chan
ges
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for deta
ils.
ttn@TTN:~/repo2$
```

17. Stash the changes and pop them

A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Tue 22:02, ttn@TTN: ~/repo1). The terminal shows the following sequence of commands and output:

```
ttn@TTN:~/repo1$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   1.txt

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

    2.txt

ttn@TTN:~/repo1$ git stash
Saved working directory and index state WIP on master: 9d2dd24 commit by repo 1
ttn@TTN:~/repo1$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    2.txt

nothing added to commit but untracked files present (use "git add" to track)
ttn@TTN:~/repo1$ git stash -u
Saved working directory and index state WIP on master: 9d2dd24 commit by repo 1
ttn@TTN:~/repo1$ git status
On branch master
nothing to commit, working tree clean
ttn@TTN:~/repo1$ git stash pop
Already up to date!
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    2.txt

nothing added to commit but untracked files present (use "git add" to track)
Dropped refs/stash@{0} (16dc1ab11dbef0b23aa2ca4d34f38b4eff2a1658)
ttn@TTN:~/repo1$ git stash pop
```

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
```

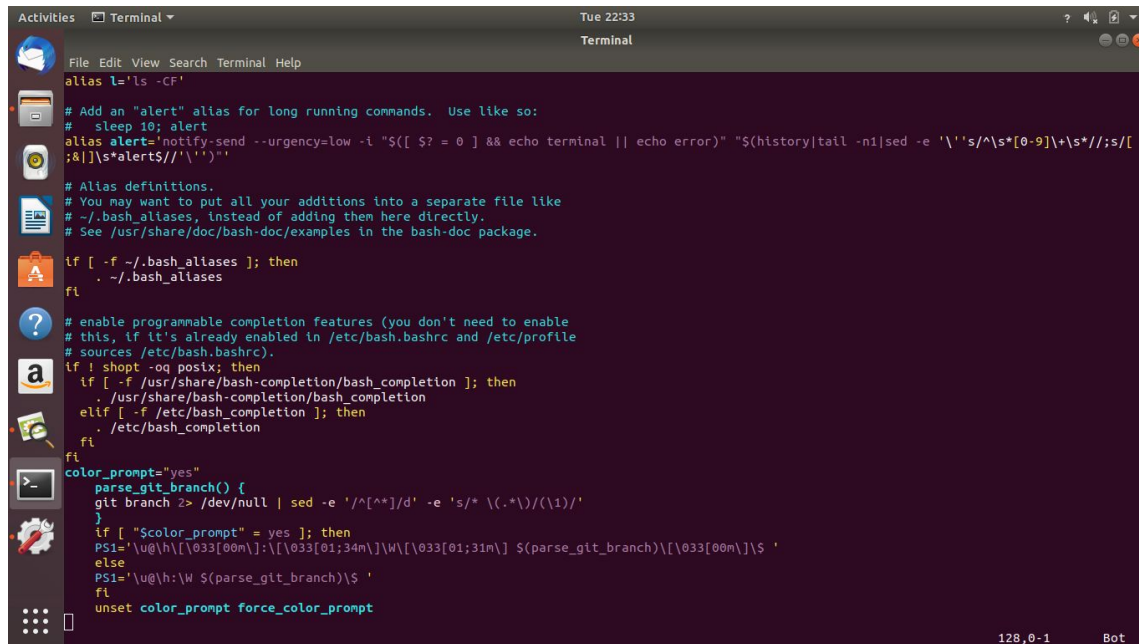
```
PS1='\u@\h\[\033[00m\]:\[\033[01;34m\]\W\[\033[01;31m\]
$(parse_git_branch)\[\033[00m\]\$ '
```

```
else
```

```
PS1='\u@\h:\W $(parse_git_branch)\$ '
```

```
fi
```

```
unset color_prompt force_color_prompt
```



```
Activities Terminal Tue 22:33
Terminal
File Edit View Search Terminal Help
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
```