Ritesh Singh
empId 3239
Introduction to Version Control

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

```
ttn@ttn:~$ sudo apt-get install git
[sudo] password for ttn:
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 430 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 20s (241 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 125727 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) ...
```

2. Initialize a Git Repository

```
                              ttn@ttn: ~
 File  Edit  View  Search  Terminal  Help
ttn@ttn:~$ git init MyTest
Initialized empty Git repository in /home/ttn/MyTest/.git/
ttn@ttn:~$
```

3. Add files to the repository

4. Unstage 1 file

5. Commit the file

```
File Edit View Search Terminal Help
ttn@ttn:~/MyTest$ git commit -m "Second commit"
[master 53506c0] Second commit
 1 file changed, 1 insertion(+), 1 deletion(-)
ttn@ttn:~/MyTest$
```

6. Add a remote


```
File Edit View Search Terminal Help
ttn@ttn:~$ git remote add origin git@github.com:RiteshSingh0097/MyTest.git
ttn@ttn:~$ git remote -v
origin  git@github.com:RiteshSingh0097/MyTest.git (fetch)
origin  git@github.com:RiteshSingh0097/MyTest.git (push)
ttn@ttn:~$
```

7. Undo changes to a particular file


```
File Edit View Search Terminal Help
ttn@ttn:~/MyTest$ 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:   README

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

8. Push changes to Github

```
File Edit View Search Terminal Help
ttn@ttn:~/MyTest$ git commit -m "final"
[master 06aa608] final
 1 file changed, 1 insertion(+), 1 deletion(-)
ttn@ttn:~/MyTest$ 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), 283 bytes | 283.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:RiteshSingh0097/MyTest.git
   14849ef..06aa608  master -> master
ttn@ttn:~/MyTest$
```

9. Clone the repository

```
File Edit View Search Terminal Help
ttn@ttn:~/MyTest$ git clone git@github.com:RiteshSingh0097/Java_Program.git
Cloning into 'Java_Program'...
remote: Enumerating objects: 11, done.
remote: Total 11 (delta 0), reused 0 (delta 0), pack-reused 11
Receiving objects: 100% (11/11), done.
ttn@ttn:~/MyTest$
```

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

-> adding changes to file

```
File  Edit  View  Search  Terminal  Help
ttn@ttn:~/My/MyTest$ 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:   sample.c

no changes added to commit (use "git add" and/or "git commit -a")
ttn@ttn:~/My/MyTest$ git commit -am "copy"
[master 9f8fa48] copy
 1 file changed, 1 insertion(+), 1 deletion(-)
ttn@ttn:~/My/MyTest$ git push origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 320 bytes | 320.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:RiteshSingh0097/MyTest.git
   803d45f..9f8fa48  master -> master
ttn@ttn:~/My/MyTest$
```

-> pulling changes in other file

```
File  Edit  View  Search  Terminal  Help
ttn@ttn:~/MyTest$ cat sample.c
#include<stdio.h>
int main(){
        printf("Hello World");
        return 0;
}
ttn@ttn:~/MyTest$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:RiteshSingh0097/MyTest
   803d45f..9f8fa48  master      -> origin/master
Updating 803d45f..9f8fa48
Fast-forward
 sample.c | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
ttn@ttn:~/MyTest$ cat sample.c
#include<stdio.h>
int main(){
        printf("Hello folks!");
        return 0;
}
ttn@ttn:~/MyTest$
```

11. Check differences between a file and its staged version

12. Ignore a few files to be checked in

13. Create a new branch.

```
           new file:    MyTest/README.md

ttn@ttn:~/MyTest$ git branch
* master
ttn@ttn:~/MyTest$ git branch brnch1
ttn@ttn:~/MyTest$ git branch
  brnch1
* master
```

14. Diverge them with commits

commit file in branch branch1


```
                          ttn@ttn: ~/MyTest

 File  Edit  View  Search  Terminal  Help
ttn@ttn:~/MyTest$ cat file
hello world
ttn@ttn:~/MyTest$ cat >> file
bro
code
ttn@ttn:~/MyTest$ cat file
hello world
bro
code
ttn@ttn:~/MyTest$ git status
On branch branch1
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:    file

no changes added to commit (use "git add" and/or "git commit -a")
ttn@ttn:~/MyTest$ git commit -am "file updated"
[branch1 43ea8d2] file updated
 1 file changed, 2 insertions(+)
ttn@ttn:~/MyTest$
```

commit file in branch master

```
File Edit View Search Terminal Help
ttn@ttn:~/MyTest$ git checkout master
Switched to branch 'master'
Your branch is behind 'origin/master' by 4 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)
ttn@ttn:~/MyTest$ cat file
hello world
ttn@ttn:~/MyTest$ cat > file
hello folks!
ttn@ttn:~/MyTest$ git commit -am "file updated"
[master a62cbcf] file updated
 1 file changed, 1 insertion(+), 1 deletion(-)
ttn@ttn:~/MyTest$ █
```

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


```
File Edit View Search Terminal Help
ttn@ttn:~/MyTest$ git branch
  branch1
* master
ttn@ttn:~/MyTest$ cat file
hello folks!
ttn@ttn:~/MyTest$ cat > file
hello world!
ttn@ttn:~/MyTest$ git commit -am "updated"
[master dcf7a51] updated
 1 file changed, 1 insertion(+), 1 deletion(-)
ttn@ttn:~/MyTest$ git checkout branch1
Switched to branch 'branch1'
ttn@ttn:~/MyTest$ cat > file
hello world!
ttn@ttn:~/MyTest$
ttn@ttn:~/MyTest$ git commit -am "updated"
[branch1 59e81ef] updated
 1 file changed, 1 insertion(+), 3 deletions(-)
```

16. Try merging and resolve merge conflicts

```
File  Edit  View  Search  Terminal  Help
ttn@ttn:~/MyTest$ git branch
  branch1
* master
ttn@ttn:~/MyTest$ cat file
hello folks!
ttn@ttn:~/MyTest$ cat > file
hello world!
ttn@ttn:~/MyTest$ git commit -am "updated"
[master dcf7a51] updated
 1 file changed, 1 insertion(+), 1 deletion(-)
ttn@ttn:~/MyTest$ git checkout branch1
Switched to branch 'branch1'
ttn@ttn:~/MyTest$ cat > file
hello world!
ttn@ttn:~/MyTest$
ttn@ttn:~/MyTest$ git commit -am "updated"
[branch1 59e81ef] updated
 1 file changed, 1 insertion(+), 3 deletions(-)
```

17. Stash the changes and pop them

 -> updating first file

-> using stash to save changes and then pull request

-> using stash pop to update changes and remove conflicts by removing command

18.
Add the following code to your .bashrc file : color_prompt="yes"
parse_git_branch() {
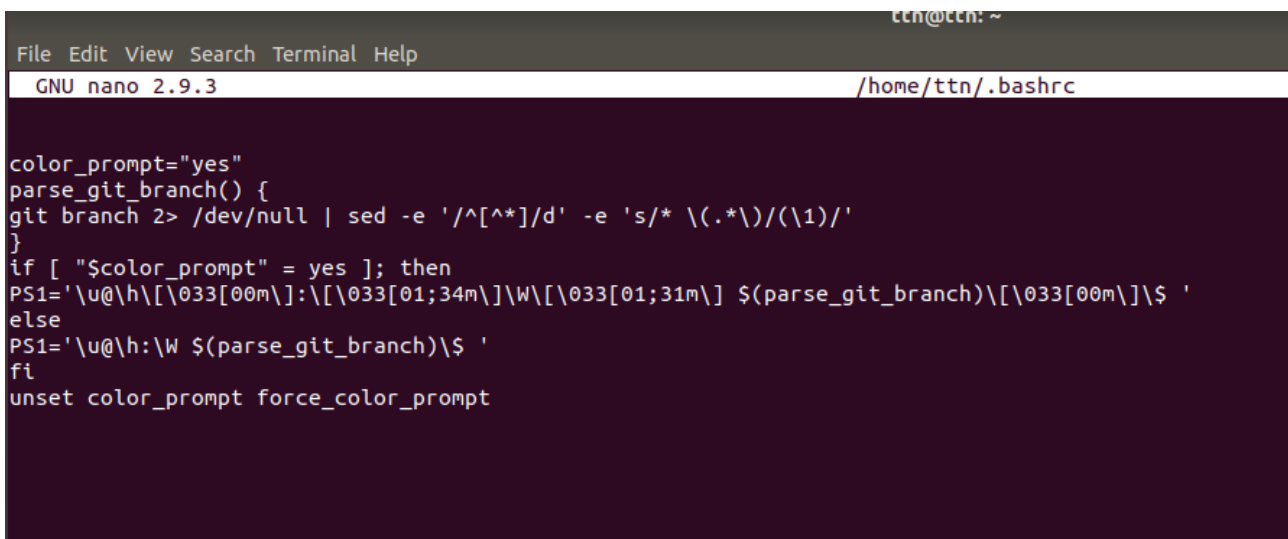git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*\)/(\1)/'
}
if ["$color_prompt" = yes]; then
PS1='\u@\h\[\033[00m\]:\[\033[01;34m\]\W\[\033[01;31m\] $(parse_git_branch)\[\033[00m\]\$ '
else
PS1='\u@\h:\W $(parse_git_branch)\$ '
fi
unset color_prompt force_color_prompt

```
                                                              ttn@ttn: ~
 File  Edit  View  Search  Terminal  Help
   GNU nano 2.9.3                                    /home/ttn/.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
```