Introduction to Version Control

Q1) Git Setup

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

Ans.

```
netik@TTNPL-netikkohli:~$ sudo apt install git
[sudo] password for netik:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.2).
```

```
netik@TTNPL-netikkohli:~$ git --version
git version 2.43.0
```

Q2) Initialize a Git Repository

Ans. To initialize the git repository we use: git init command.

```
netik@TTNPL-netikkohli:~$ mkdir demodir
netik@TTNPL-netikkohli:~$ cd demodir/
netik@TTNPL-netikkohli:~/demodir$ 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/netik/demodir/.git/
```

In above commands first demodir is created and after cd (change directory) git init is used to initialize the git repository.

A .git folder is created after executing the git init command.

```
netik@TTNPL-netikkohli:~/demodir$ ls -al
total 12
drwxrwxr-x  3 netik netik 4096 Jan 23 17:52 .
drwxr-x--- 20 netik netik 4096 Jan 23 17:38 ..
drwxrwxr-x  8 netik netik 4096 Jan 23 17:51 .git
```

Q3) Add files to the repository

Ans. git add . or git add file_name is used to add the files to the repository

```
netik@TTNPL-netikkohli:~/demodir$ touch file1.java
netik@TTNPL-netikkohli:~/demodir$ ls
file1.java
netik@TTNPL-netikkohli:~/demodir$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file1.java

nothing added to commit but untracked files present (use "git add" to track)
netik@TTNPL-netikkohli:~/demodir$ git add .
netik@TTNPL-netikkohli:~/demodir$ git status
On branch master

No commits yet

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

Q4) Unstage 1 file

Ans. The command to unstage MyFile.java is:

```
netik@TTNPL-netikkohli:~/demodir/Netik-Kohli$ 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:   MyFile.java
        new file:   file1.java

netik@TTNPL-netikkohli:~/demodir/Netik-Kohli$ git restore --staged file1.java
netik@TTNPL-netikkohli:~/demodir/Netik-Kohli$ 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:   MyFile.java

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file1.java
```

Q5) Commit the file

Ans. git commit command is used to commit and -m flag is used to give the message to the commit, which makes it easy to identify the commit in future. The message written in the quotes should be meaningful.

```
netik@TTNPL-netikkohli:~/demodir$ git commit -m "Initial Commit - Create MyFile.java"
[master (root-commit) b5de44f] Initial Commit - Create MyFile.java
 1 file changed, 2 insertions(+)
 create mode 100644 MyFile.java
```

Q6) Add a remote

Ans. To add a remote we use git remote add command :

```
netik@TTNPL-netikkohli:~/demodir/Netik-Kohli$ git remote add origin git@github.com:netik-kohli-ttn/Demo
netik@TTNPL-netikkohli:~/demodir/Netik-Kohli$ git remote -v
origin  git@github.com:netik-kohli-ttn/Demo (fetch)
origin  git@github.com:netik-kohli-ttn/Demo (push)
```

-v : Verbose used to list the remote repository

Q7) Undo changes to a particular file

Ans. Undo changes can be done by using :

```
netik@TTNPL-netikkohli:~$ git restore newFile.txt
```

Or we can use checkout with - - also to do the same:

```
netik@TTNPL-netikkohli:~$ git checkout -- newFile.txt
```

Q8) Push changes to Github

Ans.

```
netik@TTNPL-netikkohli:~/demodir$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 247 bytes | 247.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/netik-kohli-ttn/Demo1/pull/new/master
remote:
To github.com:netik-kohli-ttn/Demo1.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Q9. Clone the repository

Ans.

```
netik@TTNPL-netikkohli:~/demodir$ git clone "git@github.com:netik-kohli-ttn/Netik-Kohli.git"
Cloning into 'Netik-Kohli'...
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.
netik@TTNPL-netikkohli:~/demodir$ ls Netik-Kohli
MyFile.java
```

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

Ans.

```
netik@TTNPL-netikkohli:~/demodir/Netik-Kohli$ vim MyFile.java
netik@TTNPL-netikkohli:~/demodir/Netik-Kohli$ git add .
netik@TTNPL-netikkohli:~/demodir/Netik-Kohli$ 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:   MyFile.java

netik@TTNPL-netikkohli:~/demodir/Netik-Kohli$ git commit -m "Modiifed MyFile.java"
[master b6ee198] Modiifed MyFile.java
 1 file changed, 1 insertion(+)
netik@TTNPL-netikkohli:~/demodir/Netik-Kohli$ git push -u origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 383 bytes | 383.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:netik-kohli-ttn/Demo
   a969721..b6ee198  master -> master
branch 'master' set up to track 'origin/master'.
```

```
netik@TTNPL-netikkohli:~/demodir/Netik-Kohli$ git checkout -b branch1
Switched to a new branch 'branch1'
netik@TTNPL-netikkohli:~/demodir/Netik-Kohli$ git pull origin master
From github.com:netik-kohli-ttn/Demo
 * branch            master     -> FETCH_HEAD
```

Q11) Check differences between a file and its staged version

Ans. To check the differences between a file and its staged version we can use:

```
etik@TTNPL-netikkohli:~/demodir$ git diff MyFile.java
```

Or we can use –staged flag to find the difference between the last commit and staging area.

Here is the command:

```
netik@TTNPL-netikkohli:~/demodir$ git diff --staged MyFile.java
```

Q12)Ignore a few files to be checked in

Ans The files names in the .gitignore file is ignored by the git, the command is:

```
netik@TTNPL-netikkohli:~$ vim .gitignore
```

```
netik@TTNPL-netikkohli: ~
*.class

~
~
~
~
~
~
~
~
:wq
```

After saving this .gitignore file all the files with extension .class are ignored by the git.

Q13) Create a new branch

```
netik@TTNPL-netikkohli:~$ git checkout -b netik_branch
Switched to a new branch 'netik_branch'
netik@TTNPL-netikkohli:~$ git status
On branch netik_branch

No commits yet

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

The above command will create a new branch with name netik_branch and switched to that branch

Q14) Diverge them with commits

```
netik@TTNPL-netikkohli:~$ echo "Hello Ji" >> newFile.txt
netik@TTNPL-netikkohli:~$ git add newFile.txt
netik@TTNPL-netikkohli:~$ git commit -m "Commit newFile in netik_branch"
[netik_branch (root-commit) ab2daab] Commit newFile in netik_branch
 1 file changed, 1 insertion(+)
 create mode 100644 newFile.txt
```

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

Ans.

```
netik@TTNPL-netikkohli:~$ git checkout -b branch2
Switched to a new branch 'branch2'
netik@TTNPL-netikkohli:~$ echo "Hello Ji" >> newFile.txt
netik@TTNPL-netikkohli:~$ git add newFile.txt
netik@TTNPL-netikkohli:~$ git commit -m "Commit newFile in branch2"
[branch2 fa7e09d] Commit newFile in branch2
 1 file changed, 1 insertion(+)
```

```
netik@TTNPL-netikkohli:~$ git checkout netik_branch
Switched to branch 'netik_branch'
netik@TTNPL-netikkohli:~$ echo "Hello Netik Kohli" >> newFile.txt
netik@TTNPL-netikkohli:~$ git add newFile.txt
netik@TTNPL-netikkohli:~$ git commit -m "Commit newFile in netik_branch"
[netik_branch a09eb49] Commit newFile in netik_branch
 1 file changed, 1 insertion(+)
```

Q16)Try merging and resolve merge conflicts

Ans.

```
netik@TTNPL-netikkohli:~$ echo "hello" >> newFile.txt
netik@TTNPL-netikkohli:~$ git add newFile.txt
netik@TTNPL-netikkohli:~$ git commit -m "First commit - newFile"
[master (root-commit) 092cdec] First commit - newFile
 1 file changed, 1 insertion(+)
 create mode 100644 newFile.txt
netik@TTNPL-netikkohli:~$ git checkout -b netik_branch
Switched to a new branch 'netik_branch'
netik@TTNPL-netikkohli:~$ echo "hello Netik" >> newFile.txt
netik@TTNPL-netikkohli:~$ git add newFile.txt
netik@TTNPL-netikkohli:~$ git commit -m "Commit - newFile in netik_branch"
[netik_branch e0d096e] Commit - newFile in netik_branch
 1 file changed, 1 insertion(+)
netik@TTNPL-netikkohli:~$ git checkout master
Switched to branch 'master'
netik@TTNPL-netikkohli:~$ echo "hello master" >> newFile.txt
netik@TTNPL-netikkohli:~$ git add newFile.txt
netik@TTNPL-netikkohli:~$ git commit -m "Commit - newFile in master"
[master 68a60bd] Commit - newFile in master
 1 file changed, 1 insertion(+)
netik@TTNPL-netikkohli:~$ git merge netik_branch
Auto-merging newFile.txt
CONFLICT (content): Merge conflict in newFile.txt
Automatic merge failed; fix conflicts and then commit the result.
netik@TTNPL-netikkohli:~$ cat newFile.txt
hello
<<<<<<< HEAD
hello master
=======
hello Netik
>>>>>>> netik_branch
netik@TTNPL-netikkohli:~$ nano newFile.txt
netik@TTNPL-netikkohli:~$ git add newFile.txt
netik@TTNPL-netikkohli:~$ git commit -m "Conflict resolved in this commit"
[master cd8b204] Conflict resolved in this commit
```

We can use nano or vim to edit the newFile.txt to resolve the conflict simply by manually edit the content in the file.

Q17)Stash the changes and pop them

Ans. in git stash command takes uncommitted changes (staged and unstaged), then saves them away for later use and git stash pop is used to reapply previously stashed changes.

```
netik@TTNPL-netikkohli:~/netik$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   myfile.java
        modified:   newFile.txt

netik@TTNPL-netikkohli:~/netik$ git stash
Saved working directory and index state WIP on master: 8065bd0 Intial Commit
netik@TTNPL-netikkohli:~/netik$ git stash pop
On branch 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:   myfile.java
        modified:   newFile.txt
```

If there is no change to save then the output will be:

```
netik@TTNPL-netikkohli:~$ git stash
No local changes to save
netik@TTNPL-netikkohli:~$ git stash pop
No stash entries found.
```

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

Ans.

```
netik@TTNPL-netikkohli:~$ nano ~/.bashrc
```

```
netik@TTNPL-netikkohli: ~
  GNU nano 6.2                    /home/netik/.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)\[>
else
  PS1='\u@\h:\W $(parse_git_branch)\$ '
fi

unset color_prompt
force_color_prompt

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

Output:

```
netik@TTNPL-netikkohli:~ (netik_branch)$
```