1.Rebasing

With the rebase command, you can take all the changes that were committed on one branch

and replay them on a different branch.

Combines branch into current local branch

Move to the branch where merge need to be done

git checkout<<BranchName>>

git rebase <<BranchtobeMerged>>

2.Tags

Tagging in GIT refers to creating specific points in the history of your repository/data.

It is usually done to mark the release points.

Two main purposes of tags are:

Make Release point on your code.

Create historic restore points.

To create a tag :

git checkout{branch name}

Create a tag with some name

git tag{tag name}

To see the created tags

git tag

Push tags to remote

git push origin {tag name}

git push --tags

Delete Tags. (locally)

git tag -d {tag name}

git tag --delete {tag name}

Delete tags from remote

git push origin -d {tag name}

git push origin --delete {tag name}

git push origin :{tag name}

3.git diff

git diff lists out the changes between your current working directory and your staging area.

Showing Differences for a Specific File or Directory

git diff file\_name

Showing differences between the current version and the last version

git diff HEAD^ HEAD

4. How to rollback the deleted untracked files

git reset --hard is a classic command in this situation but it will only discard changes in tracked files.

To get rid of new (or) untracked files we use the command

git clean

5.git diff HEAD ~

Git diff is a command-line utility. It's a multiuse Git command. When it is executed,

it runs a diff function on Git data sources. These data sources can be files, branches, commits, and

more.

It is used to show changes between commits, commit, and working tree, etc.

To track the changes of this file, run the git diff command with HEAD argument.

$ git diff HEAD