**Here are the important Git commands :--**

1. **git config**
2. **git init**
3. **git clone**
4. **git add**
5. **git commit**
6. **git diff**
7. **git reset**
8. **git status**
9. **git rm**
10. **git log**
11. **git show**
12. **git tag**
13. **git branch**
14. **git checkout**
15. **git merge**
16. **git remote**
17. **git push**
18. **git pull**
19. **git stash**
20. **git config**

This command sets the author name and email address respectively to be used with your commits.

**Usage:** **git config -global user.name “Ankur”**

**Usage:** **git config -global user.email “**[**ankur@gmail.com**](mailto:ankur@gmail.com)**”**

1. **git init**

This command is used to start a new repository

**Usage :** **git init [repository name]**

1. **git clone**

This command is used to obtain a repository from an existing URL.

**Usage :** **git clone [URL]**

1. **git add**

This command adds one or more to the staging area.

**Usage :** **git add \***

1. **git commit**

This command records or snapshots the file permanently in the version history.

**Usage :** **git commit -m “[Type the commit message]”**

**Usage :** **git commit -a** (This command commits any files you have added with the git add command and also commits any files you have changed since then.)

1. **git diff**

This command shows the file differences which are not yet staged.

**Usage :** **git diff**

**Usage : git diff –staged** (This command shows the differences between the files in the staging area and the latest version present.)

**Usage :** **git diff [first branch] [second branch]** (This command shows the differences between the two branches mentioned.)

1. **git reset**

This command unstages the file, but it preserves the file contents.

**Usage :** **git reset [file]**

**Usage :** **git reset [commit hash code]** (This command undoes all the commits after the specified commit and preserves the changes locally.)

**Usage :** **git reset -hard [commit hash code]** (This command discards all the history and goes back to the specified commit.)

1. **git status**

This command lists all the files that have to be commited.

**Usage :** **git status**

1. **git rm**

This command deletes the file from your working directory and stages the deletion.

**Usage :** **git rm [file name]**

1. **git log**

This command is used to list the version history for the current branch.

**Usage :** **git log**

**Usage :** **git log -follow[file]** (This command lists version history for a file, including the renaming of files also.)

1. **git show**

This command shows the metadata and content changes of the specified commit.

**Usage :** **git show [commit hash code]**

1. **git tag**

This command is used to give tags to the specified commit

**Usage :** **git tag [commit ID]**

1. **git branch**

This command lists all the local branches in the current repository.

**Usage :** **git branch**

**Usage :** **git branch [branch name]** (This command creates a new branch.)

**Usage :** **git branch -d [branch name]** (This command deletes the feature branch.)

1. **git checkout**

This command is used to switch from one branch to another.

**Usage :** **git checkout [branch name]**

**Usage :** **git checkout -b [branch name]** (This command creates a new branch and also switches to it.)

1. **git merge**

This command merges the specified branch’s history into the current branch.

**Usage :** **git merge [branch name]**

1. **git remote**

This command is used to connect your local repository to the remote server.

**Usage : git remote add [variable name] [Remote server Link]**

1. **git push**

This command sends the committed changes of master branch to your remote repository.

**Usage :** **git push [variable name] master**

**Usage :** **git push [variable name] [branch]** (This command sends the branch commits to your remote repository.)

**Usage :** **git push -all [variable name]** (This command pushes all branches to your remote repository.)

**Usage :** **git push [variable name]** :[branch name] (This command deletes a branch on your remote repository.)

1. **git pull**

This command fetches all merges changes on the remote server to your working directory.

**Usage :** **git pull [Repository Link]**

1. **git stash**

This command temporarily stores all the modified tracked files.

**Usage :** **git stash save**

**Usage :** **git stash pop** (This command restores the most recently stashed files.)

**Usage :** **git stash list** (This command lists all the stashed changesets.)

**Usage :** **git stash drop** (This command discards the most recently stashed changes.)