**Installing Git:**

sudo apt-get update

sudo apt-get install Git

**Verifying the installation:**

git --version

**Change the defaultbranch name of the Main from Master use:**

git config --global init.defaultBranch main

**Configuring the username and email ID:**

git config --global --list

git config --global user.name "Username"

git config --global user.email "Email"

**Connecting machine to Git-Hub through SSH public key:**

ssh -T git@github.com

**Initialize repo:**

git init

**Move the file from untracked/unstaged to staged area:**

single file= git add <file name>

multiple files= git add .

**Commit :**

New file= git commit -m "message"

modified file= git commit -am "message"

single file committing= git commit --only <filename> -m "message"

**Remote Repo:**

Connect Local repo with Remote repo: git remote add origin <repo url>

Verify remote connections: git remote -v

Remove remote repo: git remote remove <name of the remote repo>

**Log commands:**

Commit list in details= git log

Commit list in one line= git log --oneline

List of all the authors within the project (s=summary and n = numbered) = git shortlog -s -n

Commits list by author= git log --author <Author Name>

OR = git log --author=<2/3 starting letters>

Logs of specific file= git log --author <Author Name> <filename>

Logs for a specific file= git log -p <fileName>

Decorated view of log= git log --all --decorate --oneline --graph

OR= git log --graph --oneline

Logs from remote repository - git log reponame/branchname

Log view of specific commit= git show <comitid> --stat

View last few commits= git log -<no of commits you want to view>

View logs between specific dates= git log --after <YYYY-MM-DD> --before <YYYY-MM-DD>

Detailed view of logs in one line= git log --stat --oneline

To read the total count of commits in the repo: git log --oneline | wc -l

**Reflog Command:**

To view the backend log with even the hidden steps we can use: git reflog

To see all the details of reflog use - git reflog show --all

**Renaming a file from git:**

git mv <old name> <new name>

**View the list of configurations:**

git config --global --list

**Alias:**

git config --global alias.br branch

git config --global alias.ci commit

git config --global alias.st status

git config --global --unset "alias.name"

git config --global --unset-all

**Diff commands:**

Difference between two commits= git diff <old commit> <new commit>

Difference between working directory and staging area: git diff

Difference between staging area and the last commit: git diff --staged

Difference between working directory and the last commit: git diff HEAD

Difference between branches: git diff <branch1 name> <branch2 name>

Difference between a commit and the staging area: git diff --cached <commit-hash>

Difference between a commit and the working directory: git diff <comit hash>

Difference between two commits= git diff <old commit> <new commit> -- filename

Modified VS Commited = git diff <filename>

Staged VS Commited = git diff --staged <Filename>

Staged VS Specific Commit = git diff <commit ID>

Remote VS Local = git diff origin/master

**Pushing Changes to remote Repo:**

1st time push= git push -u origin <remote branch>

multiple times push= git push

pushing changes to a specific branch= git push origin branchname

**Cloning a repo:**

Complete repo= git clone <URL of the repo>

Only a branch from the repo= git clone -b <Branch name><Repository URL>

**View List of Files**

List all files – git ls-files

List stagged files – git ls-files --stage

List of file in a directory – git ls-files --directory <dir name>

Configure git ignore to your system

git config --global core.excludesFile <PATH TO .GITGNORE FILE>

**Checkout Commands:**

To switch to a branch = git checkout <branchname>

To switch to a remote branch = git checkout <remotebranch>

To switch to a commit git checkout commitid

To switch to a tag git checkout tagname

**Clean Command:**

To view the implication of the clean command - Git clean -–dry-run

To force delete the file from the repo - Git clean -f

To delete dir - Git clean -f -d

To delete all the files and dir - Git clean -f -d

To delete all the ignored file in one go - Git clean -f -x

To delete all the files and dir incliding ignored files - Git clean -f -d -x

To delete a specific file - git clean -i >> option 4 >> use yes or no for selecting right file for deletion.

**RM Command:**

Removes the file from the git and the file system - git rm -f <filename>

Removes the file only from git repo but actual file will still be there in the project – git rm --cached <filename>

**Reset Command:**

Move the file to Staging area= git reset --soft <commit-id>

Move the file to unstaged area= git reset --mixed <commit-id>

Move the files out of records or reverts the file to the stage of previous commited stage git reset --hard <commit-id>

**Revert Command:**

git revert <commitID>

Note:- when you want to revert from 5th commit to 3rd commit then pass the commit id of 4th Commit to revert to the 3rd commit stage.

**Amend Command:**

git commit --amend -m “message”

**Tag Command:**

Annotated tag= git tag -a <tagname> <commit-id> -m "Message here"

Lightweight tag= git tag <tag name> <commit-id>

Get the list of tags= git tag --list

Get the list of tags with their messages= git tag -n

Delete a tag= git tag -d <tagname>

View a specific tag= git show <tagname>

Pushing tags to remote repo= git push --tags

Delete tag from remote= git push --delete origin <tagname>

**Stashing Commands:**

To stash all the files – git stash

To show the list of files stashed – git stash list

To show the latest stash or files stashed at 0th position – git stash show

To show the stashed records at position 1 – git stash show {1} or use git stash show <stash@{0}>.

To stash with a message – Git stash save “message”

To get the files back to work – Git stash pop

To get back the stashed files at position 1 – Git stash pop {1}

This will remove the files from the position 1 and it will also delete the change completely and u cannot undo this – Git stash drop {1} / {stash-id}

Remove all the stashed files in one go – Git stash clear

Stash Show more details= git stash show -p

Stashing a single file= git stash -- <filename>

**Git Branching:**

I want to see the branches available in this repository then: Git branch

I want to see all the branches including local and remote branches: Git branch -a

I want to see all the branches and their last/latest commits along with their message: Git branch -v

Delete a branch: Git branch <branchname> -d

Rename a branch: Git branch -m <oldName> <newName>

If you are on the branch you want to rename: git branch -m new-name

To check the parent of a branch = git log -g <branch name>

To view only the remote branch list: git branch -r

To create a branch and move inside the newly created branch in one go: git checkout -b <branch name>

**Git Merging:**

git merge

**git rebase:**

git rebase

**Change the default branch:**

git push --set-upstream origin <branchname>

**Shallow and Unshallow Commands:**

To clone specific amount of commits= git clone --depth <number of commits you want to clone> <project url>

To donwload more depth of commits= git fetch --deepen <no of commits you want to fetch till>

To download full depth of commits= git fetch --unshallow

Hooks