DevOps

**Git Basic Commands**

**Getting & Creating Projects**

| **Command** | **Description** |
| --- | --- |
| git init | Initialize a local Git repository |
| git clone ssh://git@github.com/[username]/[repository-name].git | Create a local copy of a remote repository |

**Basic Snapshotting**

| **Command** | **Description** |
| --- | --- |
| git status | Check status |
| git add [file-name.txt] | Add a file to the staging area |
| git add -A | Add all new and changed files to the staging area |
| git commit -m "[commit message]" | Commit changes |
| git rm -r [file-name.txt] | Remove a file (or folder) |

**Branching & Merging**

| **Command** | **Description** |
| --- | --- |
| git branch | List branches (the asterisk denotes the current branch) |
| git branch -a | List all branches (local and remote) |
| git branch [branch name] | Create a new branch |
| git branch -d [branch name] | Delete a branch |
| git push origin --delete [branch name] | Delete a remote branch |
| git checkout -b [branch name] | Create a new branch and switch to it |
| git checkout -b [branch name] origin/[branch name] | Clone a remote branch and switch to it |
| git branch -m [old branch name] [new branch name] | Rename a local branch |
| git checkout [branch name] | Switch to a branch |
| git checkout - | Switch to the branch last checked out |
| git checkout -- [file-name.txt] | Discard changes to a file |
| git merge [branch name] | Merge a branch into the active branch |
| git merge [source branch] [target branch] | Merge a branch into a target branch |
| git stash | Stash changes in a dirty working directory |
| git stash pop | Apply the last stashed state on the working directory |
| git stash apply | Like git stash pop, but do not remove last state from stash list. |
| git stash clear | Remove all stashed entries |

**Sharing & Updating Projects**

| **Command** | **Description** |
| --- | --- |
| git push origin [branch name] | Push a branch to your remote repository |
| git push -u origin [branch name] | Push changes to remote repository (and remember the branch) |
| git push | Push changes to remote repository (remembered branch) |
| git push origin --delete [branch name] | Delete a remote branch |
| git pull | Update local repository to the newest commit |
| git pull origin [branch name] | Pull changes from remote repository |
| git remote add origin ssh://git@github.com/[username]/[repository-name].git | Add a remote repository |
| git remote set-url origin ssh://git@github.com/[username]/[repository-name].git | Set a repository's origin branch to SSH |

**Inspection & Comparison**

| **Command** | **Description** |
| --- | --- |
| git log | View changes |
| git log --summary | View changes (detailed) |
| git log --oneline | View changes (briefly) |
| git diff [source branch] [target branch] | Preview changes before merging |

**Tagging**

| **Command** | **Description** |
| --- | --- |
| git tag | List existing tags |
| git tag -a [tag name] -m [tag description] | Creating an annotated tag |
| git push origin [tag name] | Sharing tags to remote server |

Basic Git Workflow

* Modify the files in your working directory.
* Move the files to the staging area by staging them – adding snapshots of them to your staging area.
* Commit, which takes the files in the staging area and stores that snapshot permanently to your Git directory.
* Git generates a unique ID which is a 40 character string of hex digits for each commit and refers to the commits by this ID rather than the version number

### Branching & Merging

* A branch in Git is lightweight and the pointer always points to one of the commits.
* The default branch name in Git is “master”.
* Each time a commit is made, the master branch pointer moves forward automatically.

## Features Of Git

**Free and open source**

**Speed**.

**Scalable**  
**Reliable**  
**Non-linear**  
**Branching**  
**Distributed**  
**Greater collaboration**

* 1. Apnagharapi
  2. Salesapi
  3. Masonenggapi
  4. Kampanel
  5. Retailcrmguiprod(already have the access)
  6. Agretail
  7. Consumerapi
  8. Retailweb
  9. Shyamsteelapi
  10. Stock\_mgmt
  11. Dealerappapi
  12. Calling\_pannel
  13. Distributorapiprod
  14. Notification\_panel
  15. Contractortappapi
  16. Distributorapi
  17. Carbrandingapi
  18. Salesdashboard
  19. Retailreport
  20. Meet\_mgmt
  21. Crone\_schedular
  22. Paintorderapi
  23. Subdealerapi
  24. Retailwebadmin
      + Working of filezilla and configuration of putty