GIT CHEAT SHEET

- ★ **WHAT IS GIT**
  1. Git is also known as **global information tracker**
  2. It is used to track the files
  3. It is a version control system and source code management
  4. It will maintain multiple version of the same file
  5. It is a platform independent
  6. It is free and open source
  7. It can handle large project effectively
  8. It saves time and developers and fetch and create pull request without switching

- ★ **GIT STAGES**

  We have three stages in git
  1. Working directory
  2. Staging area
  3. Repository

1. WORKING DIRECTORY :
   - → in this stage git is only aware of having files in the project
   - → It will not track files untill we commit those files

2. STAGING AREA :
   - → The staging area is like a rough draft space, it's where you can add the version of a file or multiple files that you want to save in your next commit
   - → In other words it is the next version of our project

3. REPOSITORY :
   - → Repository in git is consider as your project folder
   - → A repository has all the project related data
   - → It contains collection of the files and also history of changes made to these files

     TYPES OF REPOSITORIES
     1. LOCAL REPO
     2. REMOTE REPO
     3. CENTRAL REPO

1. LOCAL REPO : the local repository is everything in your .git directory mainly what you will see in your local repository are all of your check points or commit. It is the area that saves everything

2. REMOTE REPO : this remote repo is a git repository that is stored on some other r remote computer

3. CENTRAL REPO : this will present in our git hub

- ★ **GIT LAB , SVN, BIT BUCKET, P4, STASH, HELIX** THESE ARE ALL ALTERNATE FOR GIT
- ★ **HOW TO INSTALL GIT :**
  Yum install git –y (yum = yellow dog updater modifier)
  Git init . Create an empty repository
  - → **Touch file name**
    Create some files in aws account
  - → **Git add file name**
    This command is used to track the files we created
  - → **Git status**
    This command is used to check whether the file is tracking or not
  - → **Git commit -m " commit message " file name**
    This command is used to commit (save) the file
  - → **Git log**
    It is used to see the git commits
  - → **Git rm --cached filename**
    It is used to untrack the file
  - → **Git add ***
    Used to track n number of files
  - → **Git add commit –m " commit message " .**
    Used to commit n number of files at a time
  - → **Git show commit id file name**
    Used to show commit file
  - → **Git log –oneline**
    To see only commit Id's and messages
  - → **Git config user.name " name "**
    For name of our author
  - → **Git config user.email " email "**
    For email of our author

GIT IGNORE

It is used to ignore the files in n number of files

(let us assume that we have 10 files we need all files except 2 files for that we will use git ignore type vim gitignore then go to insert mode and type the files which u dont want and escape from insert mode then save and exit from vim and now track ur files (git add *) then see the git status we didnt see our git ignore files)

Note : git ignore is applied only before our files is in untrack position

GIT BRANCH

Mostly branches are used for 2 main reasons

→ Multiple developers will use the code and share their data
→ Second use of branch is version control system

COMMANDS FOR GIT BRANCH

→ **Git branch** : default branch here master will be our default branch
→ **Git branch name** : it create a new branch
→ **Git checkout branch name** : used to go our created branch
→ **Git branch –d name** : used to delete our branch
→ **Git checkout –b name** : used to create and go to thatt branch at a tym
→ **Git branch –m source destination** : to rename our git branch
→ **Git branch –d branch1 branch2** : to remove multiple branches at a tym

GIT MERGE

Git merge is used to get the files from one branch to another branch
Command : git merge branch name

GIT CHERRY-PICK

Git cherry pick is used to drag only one from one branch to another branch
Command : git cheerypick commit id

GIT MERGE- CONFLICT

Git merge conflicts is used to combine both files data

Git show commitid --stat

We will see entire details of a file like autor, date, time, filename, insertion & change deletions etc.

Git commit -- amend -m " new commit message "

To change our commit message

Git commit -- amend – no –edit

Used to commit the left over file to that same logs

Git reset : used to undo the changes

For suppose if i want to delete top 2 commits then i will copy the 3rd commit id and then enter the comand of git reset and paste the 3rd commit id then see git log the top 2 commits has deleted

Git reset commitid

Git update-ref -d HEAD

To delete all commits

Git commit –amend –author "dharani<dharani@gmail.com> "

Used to put author and mail for ur latest commit

Git rebase : git rebase branch name

As same as meerge concept

Git stash : used for temporarly deletion purpose of a file in a branch ,it has to be in tarcking position without commit

git stash
git stash apply
commit id
Git stash apply

Used to see how many stashes we have applied so far

Git stash pop

To delete lastest stash (or) to delete top stash

Git stash clear

To delete all stashes

Git revert :used to undo the specific commit

Git revert commit id

Git restore : to getback deleted files

Git restore filename

Git reset –hard HEAD~

Delete files and commits

GIT HUB

- → Git hub is an web based centralised platform
- → Multiple developers will upload their individual code in git hub
- → Then finally we will implement ci/cd with the help of git hub

How to create an git hub account

- → Go to browser search git hub .com then sign in
- → Create a new account
- → Mail id , create password and user name
- → Type y and click continue
- → Then type otp which uh have on ur mail or texted msg
- → In git hub default branch is main

How a create a repository in git hub

Go to profile ------>go to your repositories ---> new --->repository name --> description --->

Public ----> add read me file --->create repository

Git remote add origin url code

Means we have to connect git with git hub repository

Git push –u origin branch name

It will ask user name and password

How to push git files in git hub repositories

- → First we have to create a git hub account

- → We have to create a git hub repository
- → Then generate a token
- → We have to create some files git and commit those files
- → Then use git push –u origin branch name
- → Then use copied url as passcode
- → It will generate an another branch in git hub

How to add local files in git hub

Go to addfiles ---> upload files ---> click on files we have to add --> then the texted files appeared in git hub repo

Git clone

Git clone is used to get files from another developer to ur repository

Git pull : git pull is used to get the changes from the git hub