Git & Git Hub Notes

* $ git init
* inititiate git in working directory
* $ git remote add origin "htts://dcdv.f"
* this will add remote repository origin to your git folder configuration
* $ git remote -v
* shows all the remote origin list(git provide 2 url one for push and one for fetch but 99% time both url same  )
* $ git show
* shows the changes in last commit
* $ git ls-files
* it will track only modified file in stagging not new file
* $ git reset HEAD README.md
* This will reset file from git stagging even if it is git added, but shows in modified
* $ git checkout --README.md
* This will reset file and remove all the changes done in file and not track ads modified(modified file if add in stagged state then it get removed)
* $ git log --oneline --graph --decorate --all
* it will provide online info, with asterik based graph(branching heirarchy), decorate give colorful branch, all will provide history of all branches
* $ git config --global alias.history "log --oneline --graph --decorate --all"
* this will create new custom command for git log with global scope and oneline,graph etc like feature inbuild
* $ git config --global --list
* This wil give us list of global commands
* $ git mv test.txt  demo.txt
* it will rename file
* $ git rm demo.txt
* remove file
* $ git add -u
* This will add updated file in stagging
* $ git add -A
* Add all file in stagging --modified as well as newly created
* $ .gitignore (\*.log this means all file with log extension ignore)
* Excluding/Ignoring File
* $ git diff last-sha/branch-name
* This will compaire last provided SHA commit with current
* $ git branch
* THis command Will give list of branch created
* $ git checkout master
* This will checkout/switch to master branch
* $ git merge branch-name
* This will be a fast forward mnerge all commits from (from)branch will be merge and head will point to curent branch(where in to merge)
* $ git branch -d branch -name
* This will delete branch
* $ git checkout -b branch-name
* This will create new branch
* $ git tag mytag
* This will create git tags
* $ git tag -d mytag
* This will delete tag
* $ git tag -a v1.0 -m "Release 1.0"
* This will add tag with note
* $ git show v1.0
* This command shows tag and date code and commit message associated with tag, help to know major milestone
* $ git stash list
* This will shows stashes created
* $ git shash
* It will save working directory and file, index(head pointer) state, save your work in progress.
* By default, git stash stores (or "stashes") the uncommitted changes (staged and unstaged files) and overlooks untracked and ignored files. Working branch and directory will be clean
* $ git stash -u or git stash --include-untracked
* stash untracked files.
* $ git stash -a or git stash --all
* stash untracked files and ignored files.
* $ git  stash save "message"
* Saved working directory and index state On master:  mmessage
* $ git stash pop
* This will pop out top index from stash stack
* $ git rest SHA --soft
* It will reset with soft
* $ git branch <new-branch-name> be64979
* This will create new branch with provided SHA
* $ git reflog
* This will give log shows commit id but reflog shows action all done in repository like merge, checkout, push etc
* $ ssh-keygen -t  rsa  -C "email\_id"
* use email id which is set as username on github , it will create id\_rsa and id\_rsa.pub file then open public key and copy all contain