git_Commands

Initial setup

1. Install git
   a) git - Downloading Package (git-scm.com)
2. Open account at github or gitlab
   a) Join gitHub · gitHub

Wrorking with git

1. git init : to initialize empty local directory, by using this command we can observe that .git (hidden) file created in that directory.
   i. To list hidden files in *cmd* use dir /a:h
   ii. To list hidden files in *powershell* use Get- childItem -Force

2. git configure : to set username and email
   a) git config –global user.name "devopsrtt"
   b) git config –global user.email "devops.rtt@gmail.com"

3. git config –global –list : to check the current user configuration
4. In git we have 3 states are there
   a) untracked files
   b) staged files
   c) committed files

5. git status : to check the status of the updated or created files or folders

6. git add . : to move files to staging area we can use individual file names or . for all files and folders in the pwd

7. git restore : to restore to previous commit

8. git reset : to reset files from staging area git reset

9. git commit -m "message" : to move files form staging area to committed files

10. git log : to check the logs (all commits)

11. git log –oneline : to get the logs in simple and one line

12. git pull : to update local repository with remote repository

13. git clone : to clone remote repository
    a) Example: git clone https://github.com/kiran-113/aws-tf-jenkins.git

14. git fetch : is used to retrieve changes from a remote repository without merging them into your local branch. It's a safe way to see what changes have occurred in the remote repository without making any changes to your current branch.

15. git merge: is used to incorporate changes from one branch into another. After running git fetch to retrieve changes from a remote repository, you can use git merge to integrate those changes into your local branch.
    a) Example git merge branch name

16. git checkout -b branch name : to create new branch
    a) This command creates new branch and switches to new branch, if you want to create new branch we can also use git branch branch name, and to switch to new branch use git checkout branch name.

17. git branch -d branch name : to delete git branch

18. git branch -r : to list all branches including remote branches

19. git remote -v : to get current working git repo

20. git branch : to list local branches

21. git switch branch name : to switch between branches

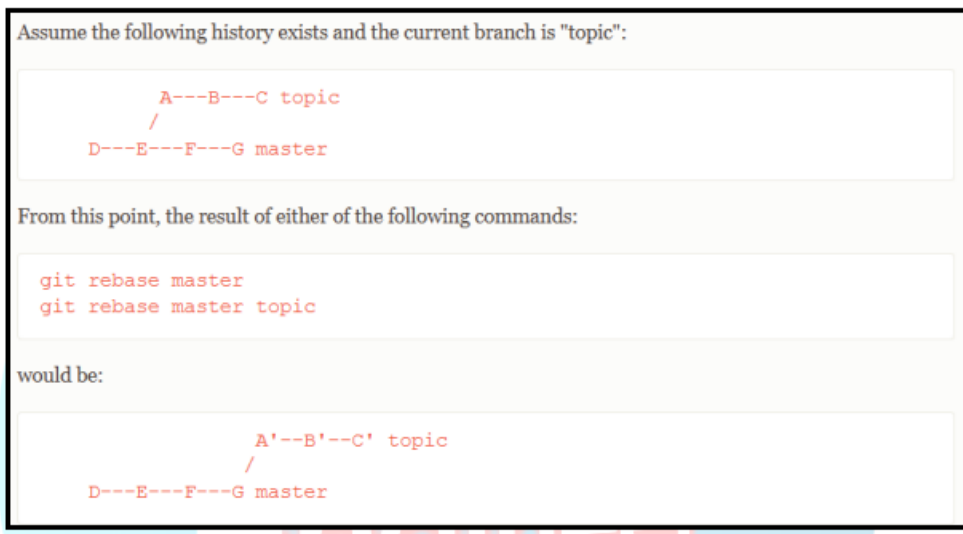
22. git cherry-pick : this command used to apply a specific commit from one branch to another. It allows you to pick a commit by using git log –oneline and apply its changes onto your current branch. This can be useful when you want to selectively merge changes from one branch into another without merging the entire branch.

23. Importance of git head : The HEAD in git is a critical and fundamental concept that plays a significant role in how git manages your version-controlled code. It's a pointer or reference to the latest commit in the currently checked-out branch

24. git reset : this command allows you to move HEAD to a different commit, effectively changing the state of the current branch.
    a) Example: git reset commit-hash # Moves HEAD to the specified commit.

25. git reset –hard : This is the most aggressive mode. It moves the branch pointer to a specific commit, resets the index (staging area) to match that commit, and discards all changes in your working directory.

26. git rebase : This command that allows you to rewrite the commit history of your branch. It does this by moving your commits onto a new base commit. This can be useful for cleaning up your commit history, or for integrating changes from another branch.



27. git stash : This command that allows you to temporarily save your uncommitted changes and then restore them later.
    a) To stash your changes, simply run the following command:
        i. git stash: will save all of your uncommitted changes to a stack
        ii. git stash list : list your stashes
        iii. git stash pop :To restore a stash, use the following command:

28. git push –set-upstream origin branch name : To push newly created branch to git , we need to do only for first time, afterwords we have to use git push command
    a) Example: git push –set-upstream origin coludsec-3423 (here cloudsec-3423 is new branch)

29. git commit –allow-empty -m "Empty commit to run build" : Is to use , when there are no changes and we need to run pipeline then we have to use this command.

30. git blame filename : Shows who what when changes happened in git repo.

31. git remote add origin repo name : used to add remote repo to local repository

a) Example : git remote add origin https://github.com/kiran-113/git_devops