Course : Devops NAME:T.Sreenivasulu

Module: git Email: srinuthippanaboina@gmail.com

Topic: git commands Assignment number: 1

Trainer name: Mr.Madhukar Date of submission: 25-11-2023

Batch number: 116

**Git commands and their use cases**

1.Git config –global user.name “name” – define the author name to be used for all commits by current user

2.Git config –global user.email “gmail” – define the author name to be used for all commits by current user

3.git clone <url from github account> – creates empty directory

4.git add . – transfer working directory to staging area

5.git status - to know the status file

6.git commit – transfer files from staging area to local repository

7.Git log – to see how many commits

8.git push – transfer files from windows to github account

9.git pull – transfer files from github to windows

10.git init – for going to master(default) branch git instalization

11.git branch – to see how many branches

12.git branch branchname – to create a one branch

13.git checkout branch – to change one branch to another branch

14.git merge branch – to merge one branch to another branch

15.git branch -d branchname – for delete the branch

16.git checkout -b branchname headnumber – for recovering the deleted branch

For head number – git reflog

17. git push origin branchname – transfer files local to remote repository

18.git diff 1st 7digits of commitid1 1st 7digits of commitid2

19.git stash save –m “stash name” - for sava data in temparory storage

20.git stash list – to see list of stashed files

21.git stash apply <stash number> – for apply stashed files

22.git stash pop – cut and paste

23.git stash drop – remove a single entry from the list of stash entrires

24.git stash clear –remove all the stash entries

25.git reset –soft commitid – 1 step back local repository to staging area

26.git reset –mixed commitid – 2 steps back local repository to staging area, staging area

to working directory

27.git reset –hard commitid –3 steps back completely delete

28.git revert HEAD <commit id> - show the changes in file in remote

29.git commit –amend –

30.git tag -a <tag name> -m <message> <commitid> - tag operation allows you giving meaningfull names to specific version in the repository

31.git tag – to see list of tags

32.git remote add <name> - creates new connection in local repository

33.merging conflict – replace the last commit with staged changes and last commit combined