

Git

```
( $\$ ) git init # First commands in git
( $\$ ) git add . or name file # all file add or one file add
( $\$ ) git add -p # git add with see which line change
( $\$ ) git ls-files
( $\$ ) git rm name file # delete file in git
( $\$ ) git rm --cached file.txt # remove git not delete local
( $\$ ) git config --global user.email 'email address'
( $\$ ) git config --global user.name 'name user'
( $\$ ) git log # for git commit log show
( $\$ ) git commit -m "message here"
( $\$ ) git commit -a -m "Message here" #This without modify fill
( $\$ ) git commit --amend #is used to make changes to commits after
( $\$ ) git revert makes a new commit which effectively rolls back
( $\$ ) git log --oneline # show one line all commit message
( $\$ ) git log --oneline branch_name # show one line only branch c
( $\$ ) git log -p Produces patch text
( $\$ ) git log --stat # show how many line add or delete
( $\$ ) git show 66e576e # Only show this commit code
( $\$ ) git diff # show difference code
( $\$ ) git diff --staged # when delete code need check which code

( $\$ ) git push origin master
( $\$ ) git push origin --delete test or branch_name
( $\$ ) git diff 6645f 435fsk # this is for check two commit diffre
( $\$ ) git checkout 6645f file name # got previous version this cor
( $\$ ) git clone 'address' # same github project name
( $\$ ) git clone 'address' myproject # this new name myproject
( $\$ ) git pull origin master #
( $\$ ) git branch # for check branch name
( $\$ ) git branch dev # create new branch
( $\$ ) git branch -d branch_nam # need merge bracnh delete
( $\$ ) git branch -D branch_nam # unmerge bracnh delete
```

```
($) git checkout -b test # create branch and switch
($) git push origin --delete dev_sorwar # dev_sorwar is branch i
($) git clone -b branch_name 'address' # git clone specific bran
($) git checkout dev # switch new banch
```

```
($) git fetch #git pull=fetch+merge and fetch=only commit messa
```

```
($) git checkout -- filename # this commands change any thing re
($) git reset filename # This commant add this git need to rem
```

```
($) git reset HEAD~1 # git reset commit lavel 1 last commit tha
($) git reset --soft HEAD~1 # git reset commit lavel 1 last comr
($) git reset --soft 456dsrw
($) git reset --hard HEAD~2 # this reset remove every where dire
```

```
($) git merge <source branch> <destination>
($) git merge test # here only source branch name bz currentlty :
```

```
($) git stash # Its work for temp un commit file
($) git stash list # show stash list
($) git show stash@{0} # change file
($) git git stash pop stash@{0} # back to stash file in directo
($) git stash apply stash@{0} # back to stash file in directo
($) git stash -p # Need partial file stash
($) git stash clean # all stash delete
($) git stash drop stash@{0}
```

Remote Repo create branch:

- > new pull request
- > comprie two branch
- > merge two branch

```
## Git global setup
git config --global user.name "Md. Sorwar Alam"
git config --global user.email "esorwar.cse@gmail.com"

## Create a new repository
git clone https://gitlab.com/sorwarduet/duet-support-service-man
cd duet-support-service-management
touch README.md
git add README.md
git commit -m "add README"
git push -u origin master

## Push an existing folder
cd existing_folder
git init
git remote add origin https://gitlab.com/sorwarduet/duet-support
git add .
git commit -m "Initial commit"
git push -u origin master

## Push an existing Git repository
cd existing_repo
git remote rename origin old-origin
git remote add origin https://gitlab.com/sorwarduet/duet-support
git push -u origin --all
git push -u origin --tags
```