**GITHUB COMMANDS TUTORIAL**

1. **Git branch**: A branch is simply pointer to specific commit.

To create git branch without switching to that branch - `git branch Hello`

To create git branch and switching to that branch at the same time - `git checkout -b hello`

To rename git local branch while being active in that branch is - `git branch -m helloWorld`

To rename git remote branch - `git branch -m helloWorld -> git push origin -u HelloWorld3`

To delete git branch locally – `git branch -D HelloWorld`

To find list of local branches - `git branch`

To find list of remote branches - `git branch -r`

To create remote branch - `git checkout -b remoteBranch -> git push origin remoteBranch`

To delete remote branch - `git push origin - - delete remoteBranch`

To delete multiple remote branch - `git push origin - - delete remoteBranch1 remoteBranch2 remoteBranch3 `

1. **Git Log**: A git log shows lists of commits(SHA) that was made to repository, each commits associated with message,commitSha,Author,date.

To check the all the git commits that made to repository - `git log`

To check most 3 recent commits that mad to repository - `git log -n 3`

To check the author who committed changes into the repository - `git log - -author=”Roni”.

To check the committer who committed into the repository - `git log - -committer=”Roni"`.

1. **Git Cherry Pick:** Git cherry pick takes changes from target commit and place them into the HEAD of the current working branch. Cherry picking is helpful if you accidentally commit the changes into wrong branch. Git cherry allows you to select specific commit from a branch where git merge/rebase selects all the commits from a branch or combined.

To cherry pick the commits from one branch to another, branch1 copy sha by using git log then switch to current working head branch and do - `git cherry-pick 2acsacs -> git push`