Name: Kuchana Rajkumar Batch no: 116

Mail id: rajkumarkuchan896@gmail.com Trainer: Mr. Madhukar Sir

Course: Devops

# Git commands and their case study

**git config --global user.name :**

**git config --global user.email :**

Configuring user name and email id in global level, i.e whenever we make commits from our local repository this username and email id are associated with your commits.

**git clone :**

First create a repository in git hub and copy the link of the repository, now use this command. This command will create a local copy of that repository which includes all the files, commit history, and branches from the remote repository.

**git status :**

it shows the status of our files like at which stage is our files are.

* + 1. if our file name is displayed in “Red” colour it means our files are at working directory stage.
    2. If our file name is displayed in “Green” colour it means our files are at staging area.
    3. If our file name is not displayed it means our changes are committed we can push to remote repository.

**git commit -m "comment for commit" :**

It moves our files which are at staging area to local repository.

**git config --list :**

This displays the credentials that you used while we are configuring the git hub.

**git push :**

This command pushes our changes from local repository to remote repository. if you made changes in only main branch then you can use this command to push files to remote repository.

**git push origin branch name :**

This command pushes our changes from local repository to remote repository.

When the changes are related to particular branch.

**git pull :**

This command pulls the files from main branch of remote repository and makes a copy of that files in local repository.

**git pull origin branch name:**

This command fetches any new changes from the remote branch and merges them into your local branch.

**git add . :**

It moves all files which we created from working directory to staging area later we can commit our changes.

**git add <file name> :**

It moves a particular file which we created from working directory to staging area later we can commit our changes.

**git log :**

It displays the total commits which are done in the local repository as well as remote repository.

**git log {file name} :**

It shows that total commits for that particular file in local repository as well as remote repository.

**git stash apply stash id:**

This command is used to bring back the data which is saved using stash id.

**git stash list :**

This displays the list of stash’s that are saved.

**git stash clear :**

It clears all the list of stash that are made.

**git stash drop :**

It removes the recent stash from the list of stash

**git stash drop {stash id} :**

It removes that particular stash from the list of stash.

**git stash pop :**

This removes the recent stash by transferring the data to that file.

**git reflog :**

It maintains a record of all the movements in the Git repository, such as commits, checkouts, merges, and resets. Essentially, it keeps a log of all the updates to the HEAD and helps you track changes even if you've lost track of them through branch deletion, resets, or other operations.

**git remote add repository link :**

The git remote add <name> <repository link> command in Git is used to add a new remote repository to your local Git environment.

**git stash save -m "comment for stash":**

This command saves the recent changes temporarily with a stash id and that recent change will not be there in file.

**git reset --soft previous commit id :**

It removes that particular commit from that list and moves changes 1 step

backward, i.e moving changes from local repo to staging area (or) staging area to working directory but data will be there in that file.

**git reset --mixed previous commit id :**

It also removes that particular commit from that list and moves changes 2 step backward, i.e moving changes from local repo to working directory but data will be there in that file.

**git reset --hard previous commit id :**

It also removes that particular commit from that list and removes data from the file.

**git revert HEAD commit id :**

This command will create a new commit that undoes the changes introduced by the commit

**git init :**

This command is to create empty git repository is specified directory. Run with no arguments to initialize the current directory as a git repository.