**GIT BASICS**

GIT is a “version control system” tool.

(or)

GIT is a “source code management” tool.

VCS(version control system):

To maintain multiple versions of a files or directories.

VCS Tools:

1.SVN

2.GIT

3.STASH

4.P4

5.CLEAR CASE

6.BIT KEEPER

* SVN and GIT are free tools.
* STASH,P4, CLEAR CASE,BIT KEEPER are paid tools.
* To track different versions of a files or directories.
* All ext based files like html, jsp, java, php , etc are tracked by vcs.

Images, videos, audio files, etc are not tracked.

* GIT is distributed version control system.GIT is developed by linus torvalds in 2005.
* GIT supports windows,mac,linux.

**DEVELOPMENT :**

1.First we need to create a directory.

Cmd :

mkdir directoryname

2. Main purpose of using GIT is to track above created directory.

3. We need to initialize the GIT.

Cmd:

git init

4. 3 phases will created in GIT:

3.Local Repo/GIT Directory

2.Staging area/Cache area

1.working directory

* GIT is aware of file in the project but not track any file in the project but not track any file.To track files,we have to commit these files by first adding the files by adding the files to the staging area.This brings us to next state in git life cycle.

Cmd:

Git add filename

* While we are in the working directory we select the files like source code file,data file,configuration file,project artifact contains business logic of application that have to tracked by GIT.

CMD:

git commit -m “commit msg”

* Now that the files to be commited are grouped and ready in the staging area.we can commit these files.

Local repo stage GIT start tracking these file now eligible to move from local repo to central repo.

FLOW OF UPLOAD DIRECTORY TO GIT :

$mkdir directoryname

$git init

$git status

$git add filename

$git commit –m “msg”

$git log

* To know the content of tracking file

$git shawn number

o/p:

All content in the file is displayed

COMMANDS:

CONFIGURATION:

#First tell git who we are:

$ git config --global user.name "Sam Smith"

(or)

$ git config --global user.email sam@example.com

INITIALISATION:

#Create a new local repository :

$ git init

CLONE :

# Create a working copy of a local repository:

$ git clone /path/to/repository

(Or)

To obtain repository from an existing url the command is:

$ git clone [url]

ADD FILE :

# Adding one or more files to staging:

$ git add filename

or

$ git add \*

COMMIT:

# commit any file you want to add to GIT

$git commit –m “msg”

#commit any file you've added with git add, and also commit any files you've changed since then:

$ git commit –a

PUSH :

#This command Sends the commited changes of the master branch to your remote repository:

$git push [variable name] origin master

This command sends the branch commits to your remote repository

$git push [variable name] [branch]

This command pushes all branches to your remote repository

$git push –all [variable name]

This command deletes a branch on your remote repository

$git push [variable name] : [branch name]

PULL:

This command fetches and merges changes on the remte server to your working directory.

$git pull remotename

LOG :

#This command is used to list the version history for the current branch.

$ git log

This command lists version history for a file,including the renaming files also.

$git log –follow [file]

FETCH :

Exactly like git pull , this command is used to download files and commits from a repository into local repo.

$ git fetch remotename

MERGE :

Merging allows developers to tocopy commits from one branch into another

$git merge[new branch name]

STATUS:

List the files you have changed and those you still need to add or commit

$git status

SEARCH:

Search the working directory

$git grep “directory”

DIFFERENCE:

#This command shows the file differences which are not yet staged.

$git diff

This command shows difference between the files in the staging area and the latest version present.

$git diff –stagged

This command shows the difference between the two branches mentioned.

$ git diff [first branch] [second branch]

RM:

#This command deletes the file from your working directory and stages the deletion.

$ git rm [file]

#This command shows the metadata and content changes of the specified commit.

SHOW:

$git show [commit]

#This command lists all the local branches in the current repository

RESET:

#This command unstages the file,but it preserves the file contents.

$git reset [file]

This command undergoes all the commits after the specified commit and preserves the changes locally.

$git reset [commit]

This command discards all history and goes back to the specified commit

$git reset –hard [commit]

SHOW

#This command shows the metadata and content changes of the specified commit.

$git show [commit]

TAG:

#This command is used to give tags to the specified commit

$git tag [commitID]

BRANCH:

#This command lists all the local branches in the current repository

$git branch

This command creates a new branch

$git branch [branch name]

This command deletes the feature branch

$git branch –d [branch name]

CHECK OUT:

#This command is used to switch from one branch to another.

$git checkout [branch name]

This command creates a new branch and also switches to it

$git checkout –b [branch name]

STASH:

#This command temporarily stores all the modified tracked files.

$git stash save

This command restores the most recently stashed files.

$git stash pop

This command list all the stashed changesets.

$gi stash list

This command discards the most recently stashed changeset.

$git stash drop