**GIT**

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency

**SCM**(Source code management tool)

GITHUB  
GITLAB

BITBUCKET

SVN>SUBVERSION

CVS>CONCURRENT VERSION

TFS>TEAM FOUNDATION SERVER

In Real time we donot use public github.we use Enterprise edition of github.

GHE>GITHUB ENTERPRISE EDITION

For example>

<https://ghe.mss.com> (GHE)

<https://github.com/mirujit(public)>

Any new project start for your company. They prepare some source code for their project .They a store those source code for that git account and inside that account they store the data.

CREATE GITHUB ACCOUNT

<https://github.com/>

>sign up >fill information

Create an organization

Click on + symbol and select new organization>follow step

<https://github.com/tech-cs-app-test> (organization URL)

<https://github.com/techpragana-batch12(GHE> or Organization URL)

Create Repository under our Organization (change repository owner name first then product name)

Choose public or private

<https://github.com/tech-cs-app-test/wallmart.git> (https)

[git@github.com:tech-cs-app-test/wallmart.git](mailto:git@github.com:tech-cs-app-test/wallmart.git) (SSH)

Create the Team

Click on organization name and click ‘Teams’ tab and click new team

Give a team name >select visible and create team .

<https://github.com/orgs/tech-cs-app-test/teams/devteam> (team URL)

Add resources/user to team

Click on team and click on ‘member’ team and add new member by adding the github username,and so on for other also can invite and other user will get mail where they have to accept.

Here we can also select particular member and change role and make as maintainer

Repository access to team

Click respository name and click on repository tab and add repository which we have created and give permission read/write>click ADD teams and Collaborators> click Invite teams or people and give permission

**GIT COMMAND**

We need CLI environment for window(git bash)and linux(yum install git to install)

Command to move to “D:/” drive

#cd /D/

git

#cd tech

#git init (initialized command betn your folder and git acc)

#ls -la

Git has some logical area

|  |  |  |
| --- | --- | --- |
| WORKING AREA | STAGING AREA | LOCAL REPO |
| Here we create file | Only tracked file | Git hub repo |
| Untracked file | Git commit -m “msg” | https://github.com/tech-cs-app-test/wallmart.git |
| Git add .()all | Git push ….. |  |
|  |  |  |

#git config --global user.name “mirujit”

#git config --global user.email [prasenjit6885@gmail.com](mailto:prasenjit6885@gmail.com)

#git config --global --list (to chk the list)

#git status(to check whether file is in untrack or tracked mode)

#git add . (all file) or git add \*.txt

#git commit -m “1st try”

git commit - - amend -m “1st try”

#git remote add wm https://github.com/tech-cs-app-test/wallmart.git

#git push wm(alias) master

**To Clone Repository: to take a copy of file in the respository.**

(go to Github click on repository hyperlink you want to clone it,Click on “code” copy the hyper link and paste it.execute the following command.

#git clone https://github.com/tech-cs-app-test/wallmart.git

PAT (personal Acess token)

First go github account setting and scroll down and click Developer setting then click personal access token>generate new token and this token (copy it),

#after this generate token key (ghp\_Cto4yxlDhja6d4WopYQvxDG8znDX8V4KIAg2)

#git log (to chk number of log)

#git log -2

#git show commit ID(To display filename in and info of particular commit)

#git show --pretty "e39ef99cb" --name-only

VCS(version control system) it is inbuit in git

To reverse from git add.

#git reset

or

#git reset filename

#git revert commit id (it will delete from local repo)

#git revert wm master (it will overwrite remote repo) but only last commit

What is difference git and github

>git is vcs

>git hub is a hosted services for the git,

#git clean -n (we can chk what we are removing from working area which in untrack area )

#git clean -f (it will delete that file)

**Git ignore file**

It will help us to ignore file which we donot want to commit.so for that we create a file .gitignore manually or while creating our repository we can add .gitignore by checking box

Example:

We will create some unwanted file and we will add those in .gitignore

#touch .classpath

#touch .settings

#mkdir node\_modules

#touch node\_modules/test.sh

#git status

#vi .gitignore

.classpath

.settings

node\_modules/ (forward slash only to identify director)

#git status

#git add

#git commit -m “3rd try”

#git push wm master

**Branching**

#git branch (it will show how many branch are there)

#git branch development (it will create branch)

#git checkout development (to move to chosen branch)

#git push wm development (to push file into particular branch)

#git branch -r (to check remote repository branch)

#git branch -a (to check both local and remote)

#git checkout master

#git merge development(it will merge file inside master branch)

#git difference development (to check diffn bet them)

#git push wm master

#git checkout -b branchname (to will create new branch and also switch new created branch)

#git branch -d branchname (to delete branch from local repo)

#git push wm :branchname (to delete branch from remote repo)

#git branch -m old branchname new branch name (to rename branch)

Differnce branch and tag

|  |  |
| --- | --- |
| Branch | Tag |
| It is mutable(can be modify) | It is an immutable(cannot be modify |
| On going development | After development |
| Git branch | Git tag |

**Tagging**

#git tag (to check how many tag)

#git tagname (to create tag)

#git tag -d tagname (to delete )

#git push wm tagname

# git push wm –tags

Tag is command to create zip,archievefile(tar.gz)

**STASH**

It us to back up **file not folder** on current directory and then we can switch to different branch and work.After finish the work on different branch we can switch to old directory and recall the back up file and work.

#vi hello.txt (old file)

This line I am adding >>>>fot git stash

Hi what r you doing.

#git status (modified will come)

#git branch

#git stash (to backup and then we can move to different branch and work)

#git branch ( to see now no modified will come)

# git stash list

Then again come back to same branch were we did git tsash

#git stash apply

#git stash drop (delete back up file)

**Git crerry-pick**

it will help to pick specific file from the commit to merger into..

#git log (to check how many commit we have done for particular branch, and from that list pick any commit ID to merge with different branch instead of all)

#git branch (to chk which branch we are)

# vi hello.txt (should be present in all branch)

Then edit in one branch and push the file to repo. Then move to next branch

#git checkout master (move to different branch)

#git log (copy the commit id of the new file)

#cat hello.txt (we will not see recent file that we pushed)

#git cherry-pick commit id (after this we can see)

#git cherry-pick --abort

How to give access to different user id in git hub for write or other permission

First go inside the repository then click on setting >click manage access and give password>then invite team or add people>give permission and add

Now user should open the mail and accept the invitation

Git pull

Git fetch

Remote Repo

Git fetch

Local Repo

Git merge

Working Copy

Git pull = git fetch + git merge

**SSH keys**

**GITHUB SSH URL**

Open the git bash

Public key 🡪id\_rsa.pub

Private key 🡪id-rsa

RSA is encryption algorithm (**Rivest, Shamir, Adleman**) it also default one

Other algorithm type are DSA

#ssh-keygen -t dsa

Public key 🡪id\_dsa.pub

Private key 🡪id-dsa

Were this public key will store (user home Directory)

Windows 🡪c:\\Users\username\.ssh

Linux🡪/home/username/.ssh

Ls -la ~/.ssh(.ssh is hidden directory) to check if rsa is there or not

#rf ef ~/.shh/id\_\* (to delete exit file of .ssh)

Now how to use ssh url

#ssh-keygen (3 time enter) 🡪first ssh key

Then configure public key in github server

#cat ~/.ssh/id\_rsa.pub (copy entire thing inside)

Opengithub>settings>ssh and GPSkeys>then click new sshkeys>give the tittle and copy and paste the key from id\_rsa.pub>add ssh keys

Then open gitbash and execute command to link with git

#ssh -T git@github.com

#git remote -v (to check which url we are using)

**Now to use ssh url**

#git remote add alias name ‘copy paste ssh url’ from code (to map)

#git remote -v (to check which url we are using)

For practice v=create one file and push through ssh url

#git push aliasname branchname (if we ssh url it is not going to ask any creditional)