Git

Date :
Page No :

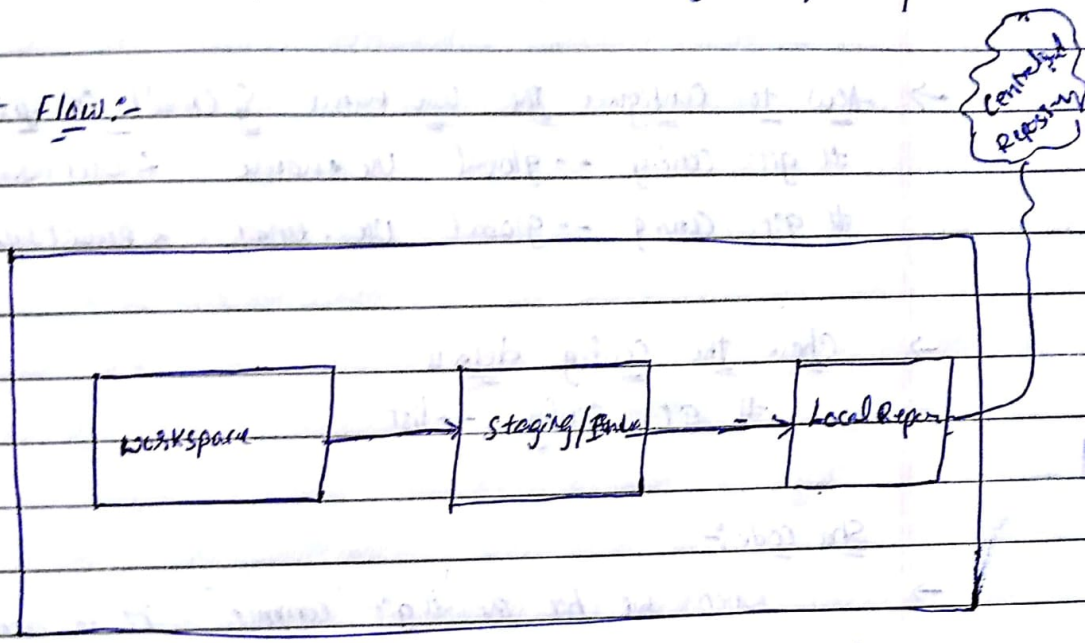
Git is a distributed version control system and source code management system.

In Git we have Three stages They are

* work space
* staging / index area
* Local Repository

* when we push the code into the Centralized Repository first we complete the workspace, staging, local Repos then we can push the code in centralized Repository.

Git Flow :-



→ To initialize the git Repository

git init

→ By default branch name is "master"

Then Create the one empty file as text file.

touch test

→ Check the files & status.

git status

↳ here if it shows the untracked file =

Then file are in workspace

(or)

If it is shows Changes to be commited files are

There in "staging area"

→ How to move the files ~~into~~ stage workspace to staging area

git add <Filename>

→ How to move the files staging area to local Repository

git commit -m "message".

→ How to Check the local Reposim file

git log.

→ How to configure the username & email in git

git config --global username <username>

git config --global user.email <email id>

→ Check the config details

git config --list

Sha code :-

→ When we hit the "# git commit" it is automatically create.

→ How to Reset the file from staging area to workspace

git reset head <filename>

→ How to Reset the file from local Repository to staging (or) Index area

git reset --soft <Commit ID> } How to see the commit ID # git log

How to Reset the file from local Repository to workspace

git reset --mixed <Commit ID>

How to see the how many files are in one sha code

git show <sha code>

How to Create Github Repository

→ Browse Github.com

↓

Sign up

↓

Create your personal account

↳ Username

email ID

password

[Create Account]

Then select public ↑

[Continue]

Then Skip and continue.

→ verify mail

↳ [Start a project]

[Repository name]

Init Readme

[Create Repository]

→ Clone : Clone is Nothing but similar copy

→ How to download the Repository from github to Localm

git clone <github Repository path>

→ How to push the code Local Repository to Centralized Rep

git push

→ Git hub.com → Create Repo (batch-20)

↓

git clone path

↓

cd batch-20

↓

touch t1 t2

↓

git Add *

↓

git commit -m "label" Filename

↓

git push

Branches:

→ How to list the branches

git branch

→ Create New branch

git branch <branch name>

→ How to change one branch to another branch

git Checkout <branch name>

Mergeing:

* First Checkout where we want to merge the code that branch Then

* # git merge <Source branch name>

* Delete the branch:

git branch -d <branch name>

→ How to create the branch and checkout at a time?

git checkout -b <branch name>

* Advantages of git? (or) Advantages of distributed system?

-> Data Redundeny and Replication

-> High availability

-> Collaboration friendly

-> Any sort of project can use Git.

-> only one git directory per repository.

* Source code Administation

(or)

Daily Activity in Git

Basicaly in version control system we are first create the Repository. after create the Repository we have a ~~paper~~ proper structure in the Repository like having following the branching strategie That is the Reasion we are Creating the Trunk, Branch, Tag That is the Repository structure Basicaly we are using Trunk we are maintain the original Source code and perticuly version. and coming to the Branch here we are Creating the multiple branches accoding to the requirement like create the bugFix branch, and then we will create the hot fix branch, standed branch and feature branch, also we create accoding to the Developer requirement once in my developent team ~~develop~~ the They will develop the Source code whatever branches are Created after that They will rise a request to create a continous Integration job apart from development and then we are using tag also in SVN. Basicaly in tag we are maintain backup of perticuly version and then once we create the package in jenkins the purpose of the tag is in feature purpose

when we required the backup I will get the code from the tag

Briefly explain about Branching?

Branching Statagie we are following Basically in SVN so TRUNK, Branch, Tag we will Create. So Trunk we are maintain the original Source Code of the perticlus version.

Comming to branchs we will Create the branches Accoding to the requirement like Standed branch, Bugfix branch, hot fix branch and also feature branch we will Create.

Basically Standed branch when we Create Before we have to ask the developer that perticlus Branch location and Name we should ask accoding that we create and we can check in the branch into our Repository:

after that any bugfix branch. Bugfix branch Nothing but wheneves issue happen into the perticlus version. So what exactly we will do is Create the Bugfix branch whateves Issue is happend That code I will push into that bugfix branch

Comming to hotfix branch. Hotfix branch is Nothing but wheneves my developes required tempory branch Suppose if they want to add Some few lines code into existing Release so That Senario we can Create the hot fix branch. but we will delete after work. They Release the code into the Application

This kind of Things we will use. and also tag we are using for taking the baekup for feature purpose. we are taking The tag.

Difference between Git and SVN ?

Main difference between SVN and Git. SVN operate with only in online but git we can use offline also. Complete repository we can download in git repository but SVN That is Not possible only folder we can check out. and also SVN their is No Reviewing area but in git we have Reviewing ~~stage~~ like stageing area. wheneves we want to check in The source code before checking we can review on The Source code in git. Comming to ~~Stat~~ branch, statagie is very easy git master branch have only in git. but in branching in SVN we are maintain differnt stages i.e Trunk, Branch, tag That statagie is difficult ~~to~~ Compare to a git. and SVN is Centralized Repository and git is distributed Repository.

Page No :

* Daily activities. work in git operations?

Basicaly in VCS we are First creating the Repository after creating the Repository we have a proper structure in the repository like here following the branching stratagie that is the reason we are creating the Trunk, branch, tag that is the repository structure. Basicaly we are using Trunk we are maintain the original source code after particular version and coming to the branch here we are creating the multiple branches according to the ~~project~~ requirement like create the bug fix branch, and then we will create the hot fix branch and similarly future branch also we will create. According to the development Requirement. once in my development team they will develop the source code what ever branches are to create after that they will rise a request to create a continous Integration Job apartform development and then we are using tag also in SVN Basicaly in tag we are maintain backup of particular version and then once we create the package in the Jenkins the purpose of the tag is in future purpose when we required the backup I will get the code from the tag so this structure we will follow. So Basicaly Branches according to the

→ SVN copy Source URL branch and dev -m

git co

→ How to move one branch to another branch in git

git co <branch name>

Git

Date :
Page No :

* What is the diff b/w git pull and git fetch?

Git pull command pulls new changes or commits from a particular branch from your central repository and updates your target branch in your local repository.

→ Git pull = git fetch + merge.

* Chef Arechitecture :-

In Chef Arschitecture have Three majos componen They are

1. Chef Serves
2. Chef Node
3. Chef workstation

→ Branching ~~start~~ strategies cesed in Devops?

* Feature branching
* Task branching
* Release branching