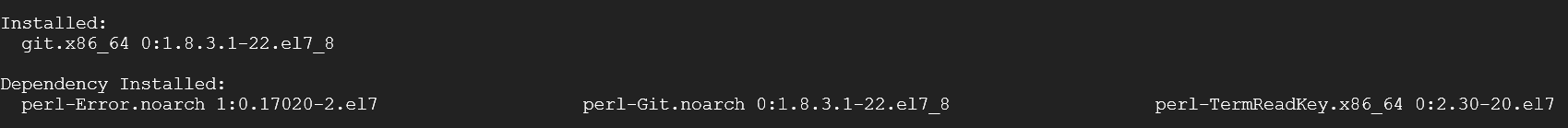
**Requirement:**

Linux Server Host with git initialization and perl dependency  


Remote server Account(Git Hub)

**Topics :**

1. Architecture & Documentation.

2. Installation

3. Git Hub Account Creation

4. Git Commands Practical Session

**Git:- Distributed Version control Tool**

* Developers can work on the same code base
* Allow multiple people to work together
* It provides the detailed log
* It provides the backup easily. Downgrade required we can able to provide
* Anyone can use like developers, testers, Admins

Scenario from company. Disapproval from client.

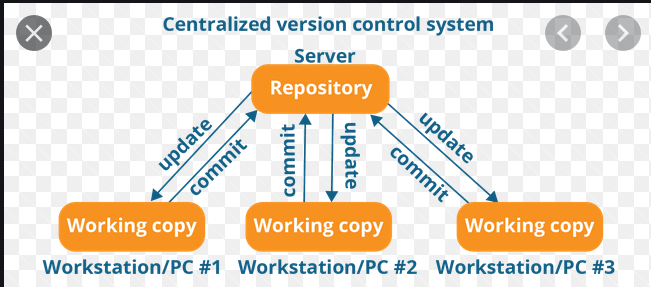
If we are not using version control, we can’t restore the removed files. For that purpose, only we are using version control



Version control is like a local database

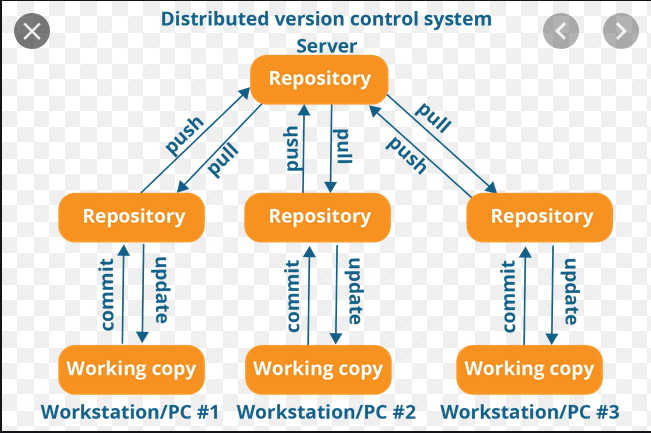
**Centralized version control system.**

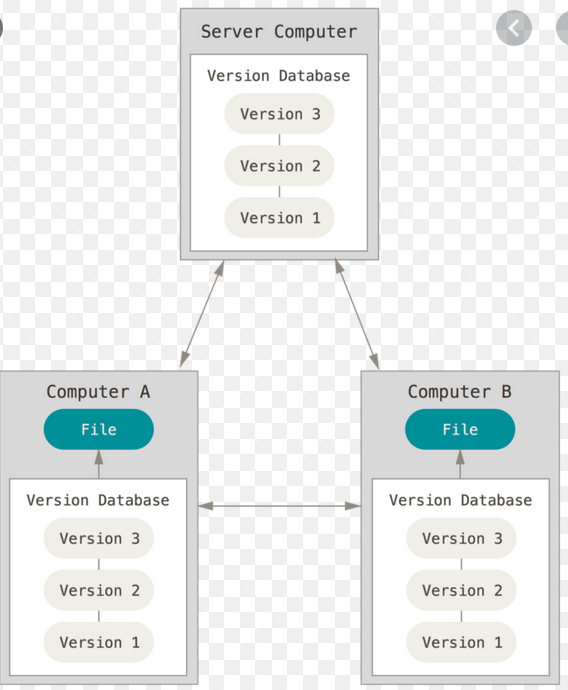
* Central repository is maintained where all the versioned files are kept. Now users can checkout and checkin files from different computers at any time.
* All program files located in central server.

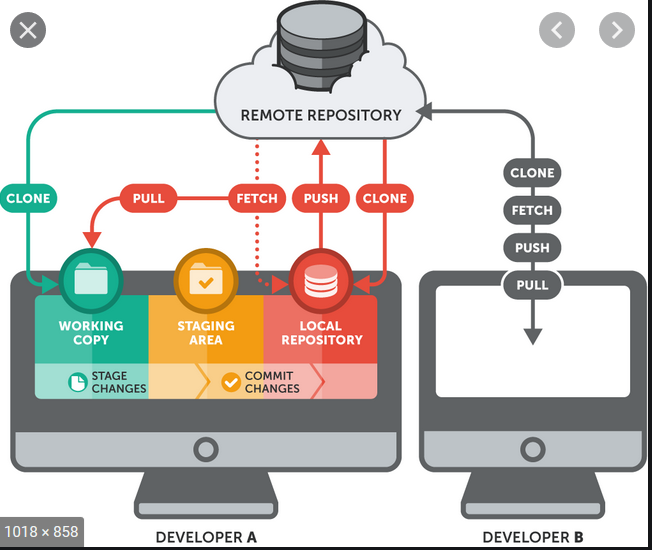
**Drawback:- The entire work will be halted if that central server went down for some issues like hardware/network issues.**

**Git Architecture:**

**To overcome that we are using Distributed Version Control Tool**





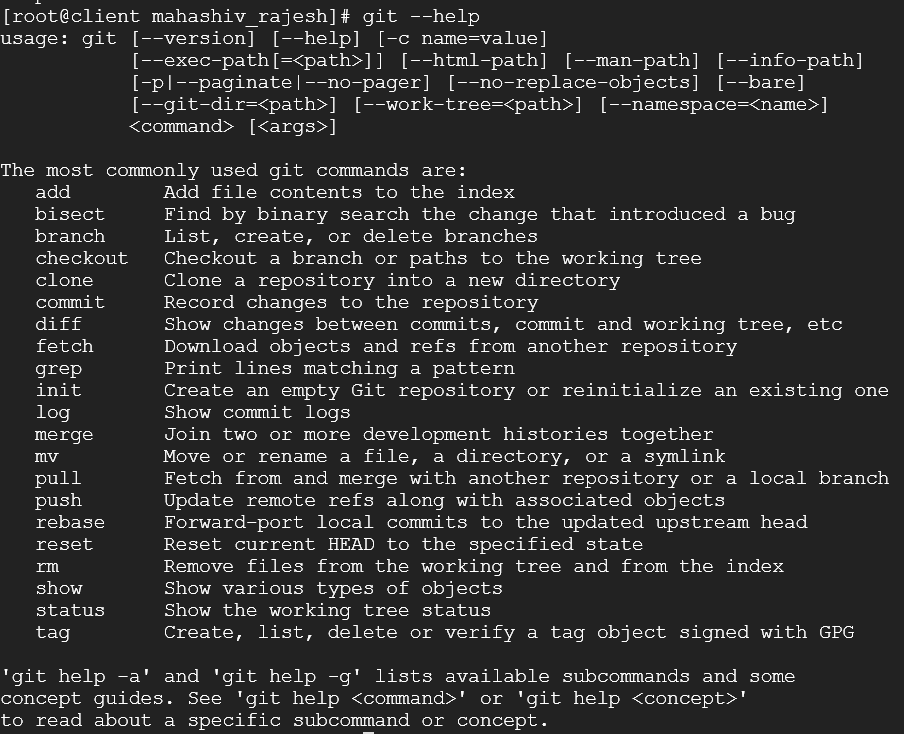


**We can create local repo in 2 ways :-**

**1 – Get a copy from remote to local computer( Anyone join in middle of the project)**

**2—If it is new project, we can able to create a repo in local machine and the we can push to remote repo.**

**Git Commands along with installation:**



[root@ansibleclient project]# git init

Initialized empty Git repository in /root/project/.git/

Local repo created in project directory

yum install git

mkdir project

cd project

git init

cat >> sample.txt

ls

git ls-files

ls -lrt

git status

git add sample.txt

git status

git rm --cached sample.txt

git status

git commit -m "first"

git add sample.txt

git commit -m "first"

git config --global user.name "prasad"

git config --global user.email prasad.cse18@gmail.com

git ls -files

git log

commit d43517c6c55d3330aa8724bcf35c317d3b337a9f

Author: root <root@ansibleclient.us-central1-a.c.driven-manifest-274807.internal>

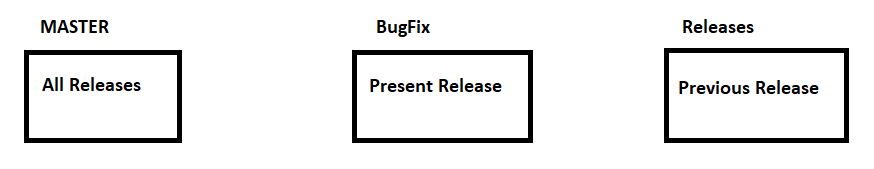
Date: Fri May 8 03:09:41 2020 +0000

First

For every commit it generates a commit id.

Branches are used to achieve a parallel / Feature development.

Like for bug fixing 1 branch and for releases another branch.

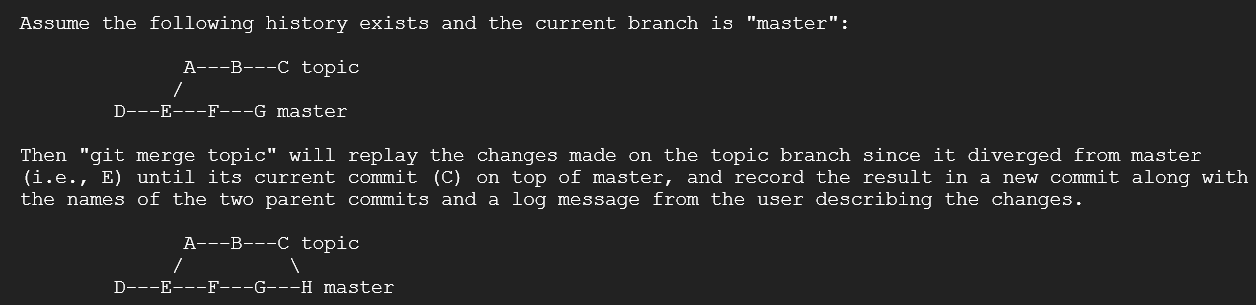


Branching allows parallel development.

Master is the main branch that contains all data.

BugFix Branch copy data from Master Branch. After committed / saved changes in the code we can once again merge with the Master branch.

Same as like Releases copying data from master add new features as per client requirement and merge with master branch



git add .

git commit -m "release commit"

git status

git log --oneline <Branchname>

git branch

git merge <Branchname>

git checkout <Branchname>

git log -help

git show --source 73a67bc

git stash -p

git stash clear

git stash pop stash{0}

git stash drop stash{0}

mahashivrajesh

Thirupati@143

git remote add origin <https://github.com/mahashivrajesh/MY_REPOSITORY>

git push origin master

It will ask credentials:

git push origin –delete <branchname>

git rm Release new sample.txt test

git reset --hard d43517c

git remote add origin https://github.com/mahashivrajesh/MY\_REPOSITORY

git push origin master

git push origin --delete master

Whenever we are pulling a new code. We must create an empty directory.

git clone <https://github.com/mahashivrajesh/MY_REPOSITORY>

when we do git clone no need to inintialize a git in that new directory.

git pull origin – it also merge the code(pull = fetch+merge)

git fetch origin – it just shows