Lesson:

Github

PW SKILLS

List of Concepts Involved:

- Git foundation
- Git software installation
- Git Architecture

Introduction

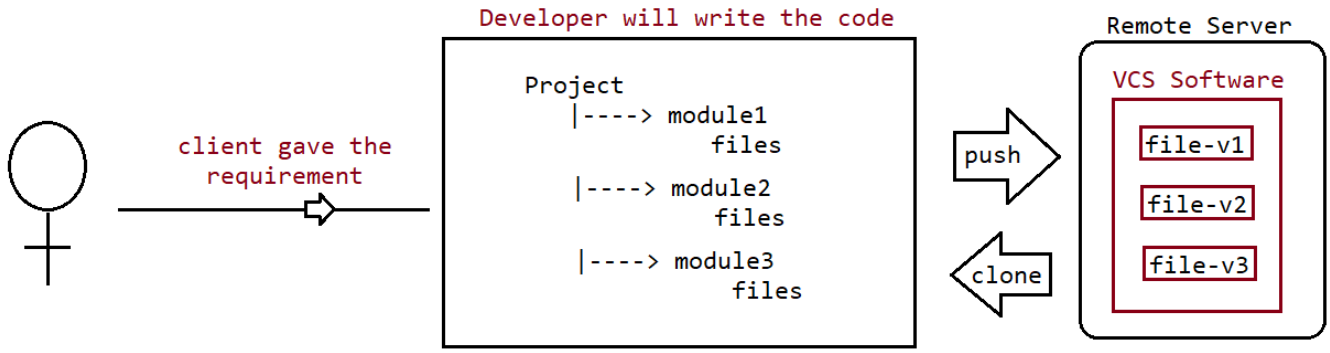
Git is a popular version control system(VCS), It was created by Linus Torvalds in 2005 and has been maintained by Junio Hamano

Git is used for

a. Tracking code changes
b. Tracking who made changes like history of the files
c. Coding Collaborations

VCS (Version Control System)

It is a system that records changes to a file or set of files over time, so that we can recall specific versions later, i.e., for every source code changes in a file a new version will be created.



There are 3 types of VCS

1. Local Version Control System (LVCS)
2. Centralised Control System (CVCS)
3. Distributed Version Control System (DVCS)

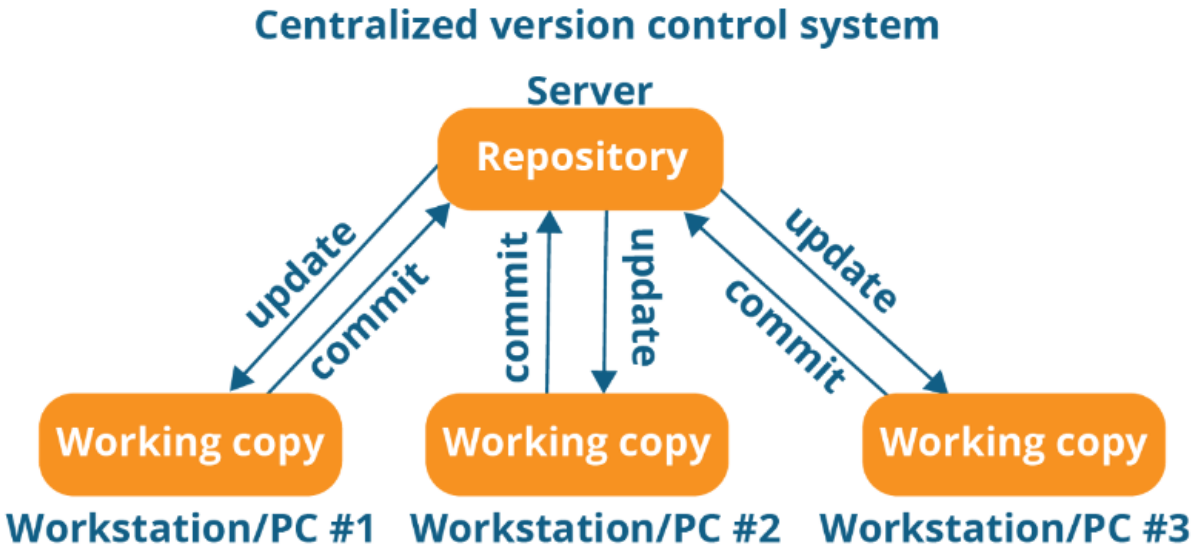
Local Version Control System

It is used to maintain the file version and retrieve the files based on specific version

Drawbacks

1. If our System is corrupted then there may be data loss issues.
2. It is easy to forget which directory you are in and some accidentally write the data to the wrong file or copy other files.
3. By mistake we can delete those files.

Centralised Version Control System
Developers can collaborate and do the changes
eg: CVS, SubVersion,Perforce



1. Centralised Version Server will have a single server that contains all the version files.
2. No more clients connect to CVCS and check out for files.
3. For Many years this has been the standard version control system.

Advantages

1. EveryOne knows to a certain degree what everyone else on the project is doing.
2. Administrators have full control over who can do what and are easier to manage.

Drawbacks

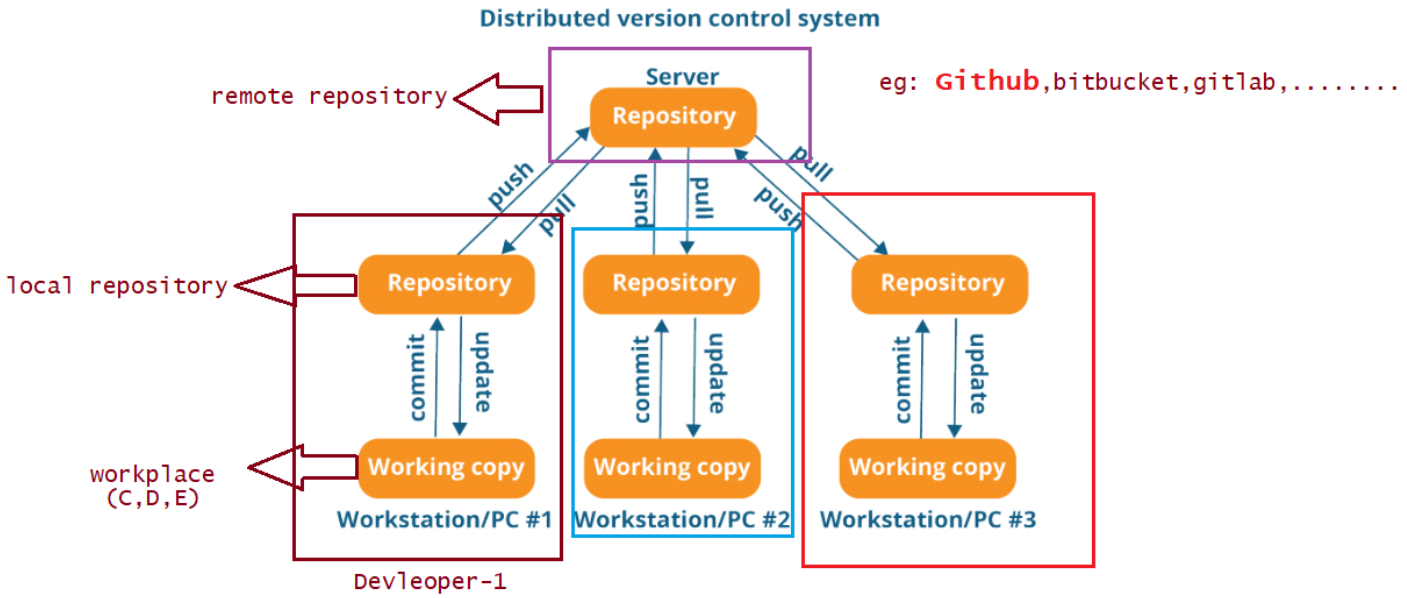
1. Single point of failure(SPF) that centralised server represents
2. If the server goes down for an hour then during that hour nobody can collaborate at all or save versioned changes to the server.
3. If the hard disk of the centralised server will be corrupted and proper backup haven't been kept then we will lose the data(source code).

Note:

1. LCVS also suffers the same problem. Whenever you have an entire history of a project in a single place, there is always the risk of losing everything.
2. CVS always gets the latest code but not the entire history of the project. This is also a major drawback of SVN.

To resolve this problem we need to use **Distributed Version Control System**

Distributed Version Control System



- DVCS are Git, Mercurial, Darcs, Bazaar,...etc
- Developers don't check out the latest snapshot of the files rather they fully mirror the repository including its full history.
- If the main server dies, then the local system will maintain a copy of the main repository which has full backup of data.
- If the remote repository is down, then the developer can make changes in the local repository and when the main repository is up the code can be pushed to remote repository from local repository.

Git Software Installation

There are 2 types of Git softwares
a. Git Server
b. Git Client

Git Server

- It is a repository
- It is the largest host of source code in the world
- It is used to store/maintain the source code of the project
- Some of the git server tools are : Github, Gitlab, BitBucket, Gitblit,....

Note:
In the case of a Developer machine we don't need GitServer we just want the url of github, username and password.

How to install Git Client software?

- It is an open source software
- It is in the form of .exe file
- Download git.exe file from https://git-scm.com/download/win
- Run the git.exe file then we will get below 3 softwares

a. git bash
Linux commands are required.

b. git GUI
Graphical user interface, developers need not enter any commands, gui will take care.

c. git cmd
Command line tools where developers can type the proper url and hit the service.

Note: using these tools we can connect to git server by providing url, username and password

Scratch Project

- Developer will develop the code in local
- move the code into remote server of branch
- do the proper testing in branch
- if it is working fine then move from branch to master
- do the proper testing in master(1.0V)
- if it is working fine then do the production deployment
- move the code from master to target , ie (release the code (version 1.0))

Git Architecture

There are 3 types in git architecture
a. working area
b. stage area
c. local repository

- Developers will develop the code in working area
- If any files want to move to a remote server then it should be from a local repository only.
- if any files want to move to local repository then it should be from stage area only, we can send the files directly from working area to local repository
- developers should develop the code from working area, then send the code to stage area,then from stage area to local
- repository and from local repository to remote repository