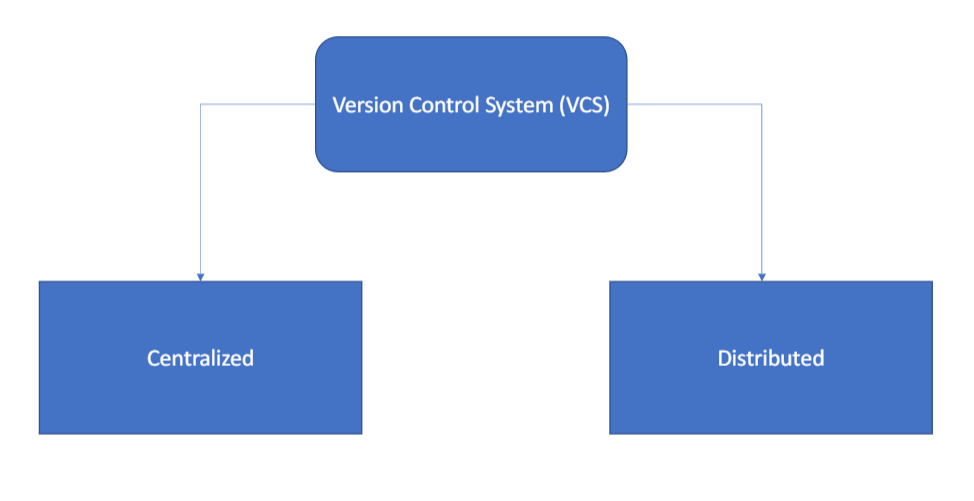
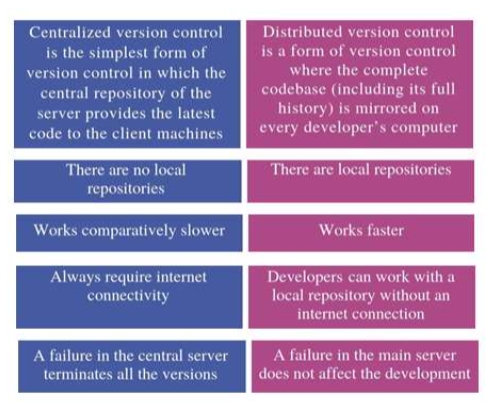
Git Hub & Version Control System

* Git is a distributed version control system (DVCS) for tracking changes to files
* Source Code Management (SCM)is the major task we need to perform during software development.
* The requirements of the software always change, and it is necessary to modify the source code
* In SCM these states of the projects are saved as versions. Therefore, the developer can keep track of the previous versions.
* Version control systems allow source code management, and it is beneficial in enterprise-level software development
* There are two types of version control systems as centralized and distributed version control systems.

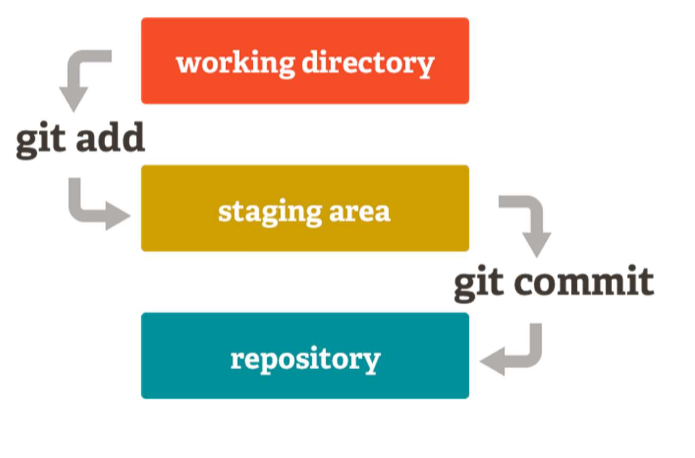


* Git is an open-source VCS, which is not file-base, unlike other systems.
* Rather it stores information as snapshot.
* Being a VCS, helps coders to revert their previous code when they hit roadblock in newer version, without affecting the original source code
* On the other hand, what makes it different from other VCS is the way it sees data, which is more like a series of snapshots.
* it basically clicks a picture of how al your files look now and saves the changes made to them over time.



Installation of git agent on windows server

* What is .git folder
* The .git folder contains all information that is necessary for the project and all information relating commits, remote repository address, etc.
* It also contains a log that stores the commit history
* This log can help you to roll back to the desired version of the code



How to create local repository by creating the folder

git init ------ Initialize a local Git repository

git config --global user.name "FIRST\_NAME LAST\_NAME" ---- Set your username

git config --global user.email “[MY\_NAME@example.com](mailto:MY_NAME@example.com)” ------ Set your email address

git config –list ---- To check all current config

git status ----- Check status

git add [file-name.txt] -------- Add a file to the staging area

git add . -------- Add all file to the staging area

git add -A ------- Add all new and changed files to the staging area

git commit -m "[commit message]" ------ Commit changes

git rm -r [file-name.txt] ----- Remove a file (or folder)

git log ----- View changes

git log –summary ------- View changes (detailed)

git diff [source branch] [target branch] ------- Preview changes before merging

How to Switch from one Commit Version to Another Version

git log ----- View changes

git add . -------- Add all file to the staging area

git commit -m "[commit message]" ------ Commit changes

git log ----- View changes

git checkout [commit number] ------ to switch from one commit to another

How to Push Code from Local Repository to Central Repository

git remote add origin <https://github.com/cloudtechnopune0610/AzureDevops.git>

https://git@github.com/[username]/[repository-name]. git

git push origin [branch name] --- Push a branch to your remote repository

git push -u origin [branch name] --- Push changes to remote repository (and remember the branch)

git push --- Push changes to remote repository (remembered branch)

How to identify commits Not pushed to Remote Repository

git log ----- View changes

git checkout [commit number] ------ to switch from one commit to another

git log ----- View changes

Working with branches

git branch --- List branches (the asterisk denotes the current branch)

git branch -a --- List all branches (local and remote)

git branch [branch name] ------ Create a new branch

git branch -d [branch name] ------ Delete a branch

git push origin --delete [branch name] ---- Delete a remote branch

git checkout -b [branch name] ------ Create a new branch and switch to it

git checkout -b [branch name] origin/[branch name] ------- Clone a remote branch and switch to it

git branch -m [old branch name] [new branch name] ------- Rename a local branch

git checkout [branch name] --- Switch to a branch

git merge [branch name] --------- Merge a branch into the active branch

git merge [source branch] [target branch] ---------- Merge a branch into a target branch

git branch -m [old branch name] [new branch name] ------- Rename a local branch

Working with pull / fetch

Git init

git config --global user.name "FIRST\_NAME LAST\_NAME" ---- Set your username

git config --global user.email “[MY\_NAME@example.com](mailto:MY_NAME@example.com)” ------ Set your email address

git remote add origin <https://github.com/cloudtechnopune0610/AzureDevops.git>

git pull https://cloudtechnopune@dev.azure.com/cloudtechnopune/mylab/\_git/mylab

git fetch ------ it will take other branches and need to use after pull

git branch

git branch -a

git checkout [branch name]

git fetch ---- It downloads latest changes into local repository. it downloads fresh changers that other

developers have pushed to the remote repository since the last fetch all allows you to review and merge manually at a later time using git merge.

because it does not change your working directory or the staging area.

git pull ----- It downloads latest changes into the local repository and it also automatically merges change in your working directory, It does not give you

to chance to review the changers before merging, and consequently, merge conflicts can and do occur.

One important point to keep in mind is that it will merge only into the current working branch, other branches will stay unaffected

git branch -avv

