Git and Github

1. Version control system tool :

A version control system is **a kind of software that helps the developer team to efficiently communicate and manage(track) all the changes that** .It is also known as software configuration management (SCM) or Source code management(SCM)

Need a version control system?

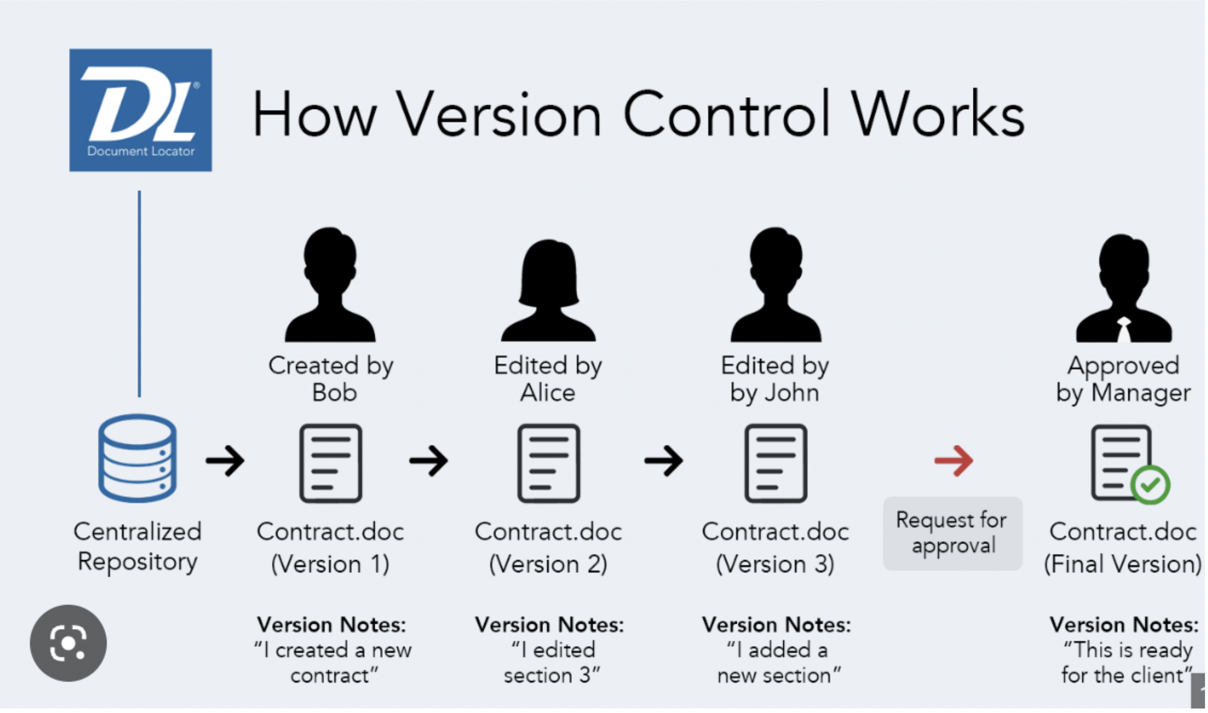
Version control software **keeps track of every modification to the code in a special kind of database**. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimising disruption to all team members.

Example –
Client name - Akash
Requirement - Develop xyz project.

DEV/QA Engineer -- Write code --Files

Client project

| - Suppose 100 java files are there
| - Client suggests some changes
| - DEV/QA changed some files from the source code to meet client requirements.
| - DEV/QA gave the demo and the client suggested some more changes.
| - DEV/QA gave the demo for the third time and the client suggested some more changes.
And at the end client says revert back all the changes and keep the same code as which was developed for the first time. That task will be difficult for us.


Document Locator
How Version Control Works
Created by Bob
Edited by Alice
Edited by by John
Approved by Manager
Centralized Repository
Contract.doc (Version 1)
Contract.doc (Version 2)
Contract.doc (Version 3)
Request for approval
Contract.doc (Final Version)
Version Notes: "I created a new contract"
Version Notes: "I edited section 3"
Version Notes: "I added a new section"
Version Notes: "This is ready for the client"

1. Every change should be tracked
   Who did changes
   When did the changes
   Which changes he did etc.

2. Overwriting the code should not happen.
3. Developers have to share their code to peer developers so that multiple developers will work in Collaborative way.
4. Parallel Development must be required

Multiple versions are required to maintain each and every change should be tracked and should be stored.
These are the need

How version control system work

Version control system always talks about files which contain source code

1. Basic version control system terminology

1. Working directory :
Where QA /Dev is required to create /modify files

where version control is not applicable, here we won't use the work like version -1, version 2, etc

Repository: Where we have to store files and metadata(Extra files)
Here version control is applicable.
Here we can talk about versions like v1,v2,v3

Commit: The process of sending files from the working directory to the repository

Checkout: The process of sending files from the repository to the working directory.

===

Benefits of version control system

1. We can maintain different versions and we can choose any version based on client requirements.
2. With every version /commit we can maintain metadata like
   Who Commit message
   Who did changes
   When did the changes
   Which changes he did etc.
3. QA engineers can share the code to the peer QA in a very easy way.
4. Multiple QA can work in a collaborative way.
5. Parallel development.
6. We can provide access control like
   who can read code
   who can modify code?

==

Git command :

Git add - sending files from the working directory to the staging area
Git commit - Staged changes will be moved to the local repository
Git Checkout - This option is performed locally via the b/w working directory and repository. (To perform this operation not required)
Git push - The process of sending files from the local repository to the remote repository
Git clone - To create a new local repository from the remote repository.
git pull - The process of getting the updated files from the remote repository to the local repository.

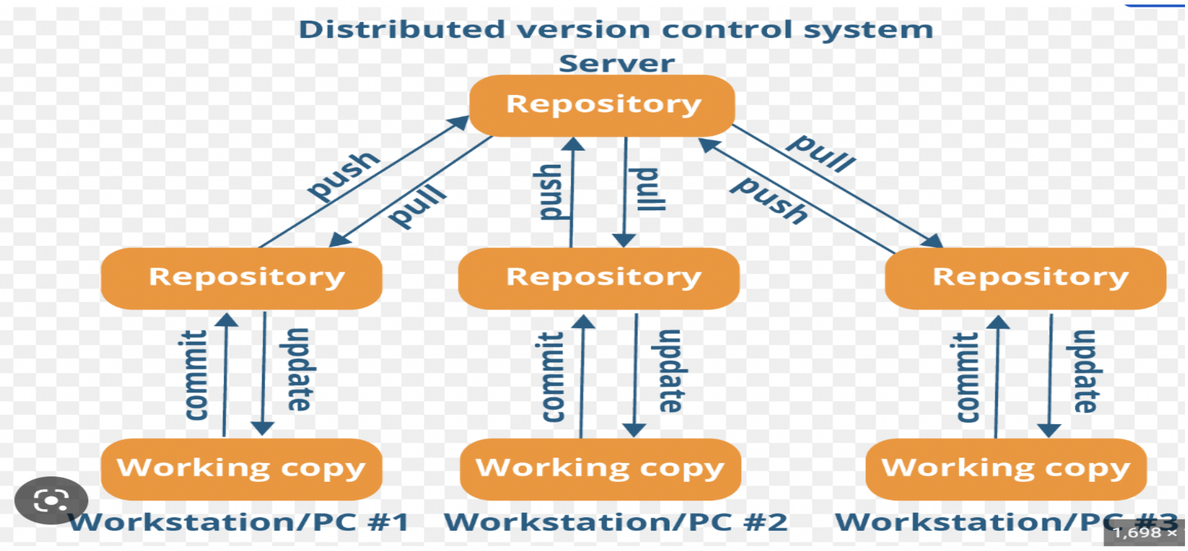
==

Remote repository server
Github
Gitlab
BitBucket
cloud perform

==

Git features

1. Distributed - In a Distributed Version Control System (such as Git, Mercurial, Bazaar or Darcs), the user has a local copy of a repository. So, the clients don't just check out the latest snapshot of the files even if they can fully mirror the repository. The local repository contains all the files and metadata present in the main repository.



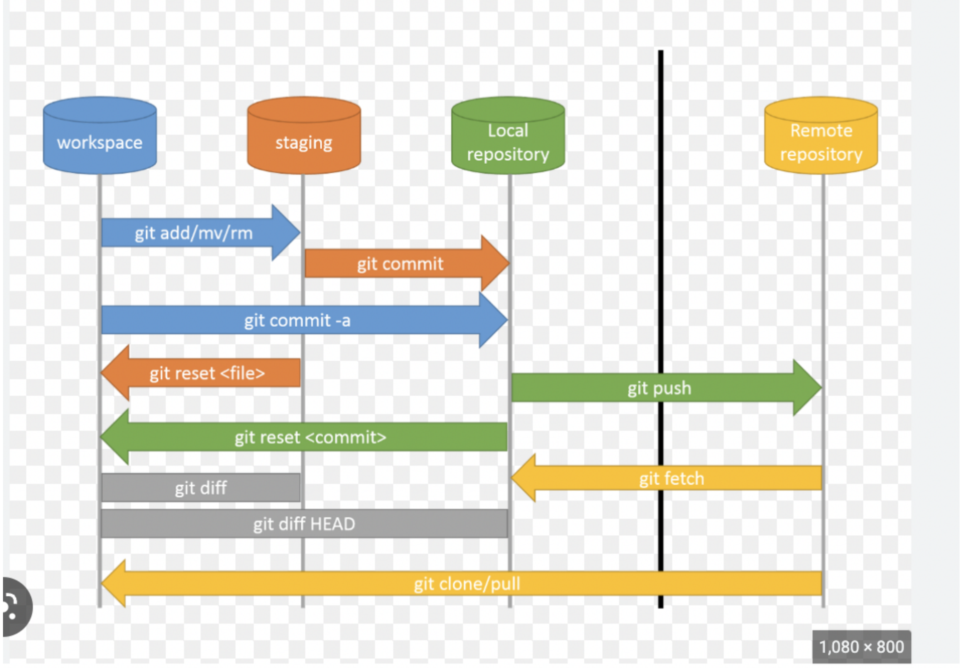
. No single point of failure .every QA has a local repository
. Performance is more /Speed more
. Without n/w also Qa can continue his work.

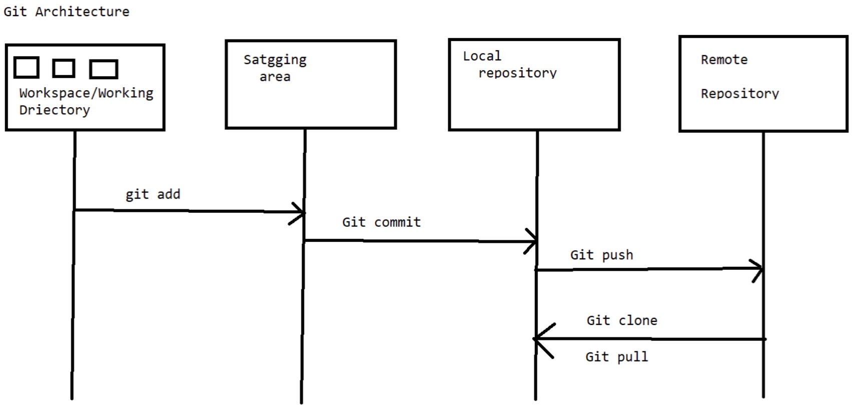
2. Staging area
. Steps process
. First, we need to add files to the staging area and then we have to commit from the staging area.
The advantage of the staging area is we can cross-check or double-check our changes before committing.if everything is fine
then we can commit.   (diagram 6)

3. Branching and merging
4. Open source
5. It provided support for multiple platforms

==

Git Architecture

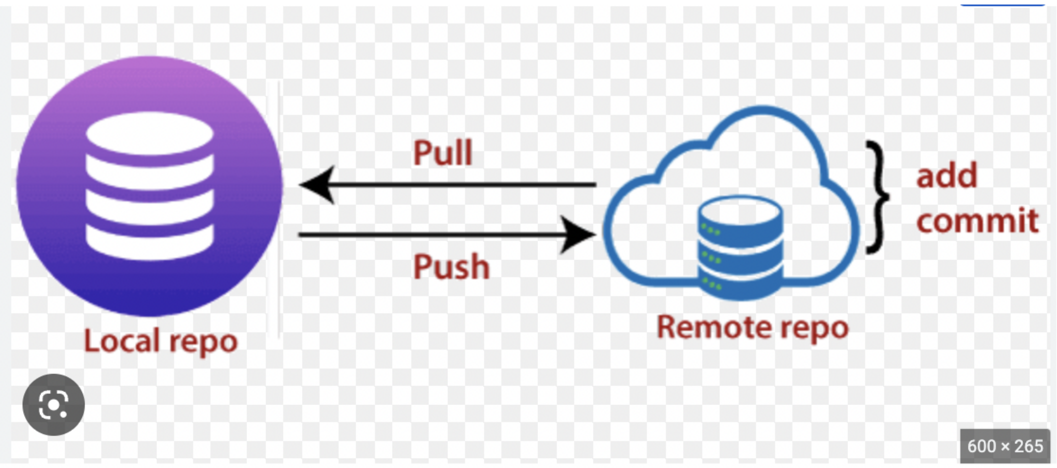




1. Staging area
2. commit
3. checkout
4. Push
5. Pull
6. remote repository
7. local repository

All these words we are using in git architecture

Git has 2 types of repository
1. Local Repository
2. Remote Repository

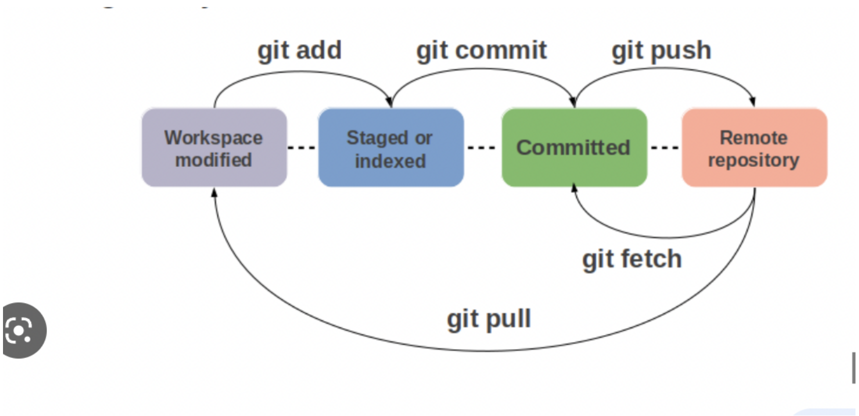


Usually, the total project code will be available in a remote repository the current work of Qa will be stored in the local repository.

The new file was created in the working directory
Once work is completed, we have to add these files to the staging area for these we have to use the git add command

Git add   - sending files from the working directory to the staging area
Git commit - Staged changes will be moved to the local repository
Git Checkout -This option is performed locally b/w the working directory and repository. (To perform this operation n/w is not required)
Git push   - The process of sending files from the local repository to the remote repository
Git clone - To create a new local repository from the remote repository.
git pull   - The process of getting the updated files from the remote repository to the local repository.

The life cycle of file in Git



1. Untracked
  Every new file will be created in the working directory. Git is not aware of these new files. such types of files
  are said to be in "Untracked".

2. Staged
The files which are added to the staging area are said to be in a staged state

3. in repository/committed
Any file which is committed is said to be in the repository state or committed state.

4.Modified

1. How to install git on window

URL - https://git-scm.com/download/win

click on 64-bit Git for Windows Setup.
it is an exe file

Question

What is Git?
Git is a version control system tool used to track changes in computer files/source code.
it's a tool to manage your code & file history while coordinating work remotely on those files with others

==

What is Github?

Github is a hosting service for git repositories.
Git is the tool while GitHub is the service to use git.

==

Branching % Merging

End-to-end process
1.
1. Create a new Project

2.Create local(git) repository
3. Add files to the staging
4. Commit changes to local Repo(.git)
5. Push changes to the master branch remote repo(Github)

2.
I created a new project on GitHub go ahead and clone it then continues with your task

3.
1. Clone remote repo to local repo(Working Directory)
2. Import project from git to workspace.
3.Add/modify feature/code/test cases
4. Add files to the staging
5. Commit changes to the local repo(git)
6. Push changes to remote repo(Github) from the new branch
7. Request PR(Pull Request) For review code

4.

Review code
Case 1- changes required then provide comments to a team member.
Case 2 - Everything is good, then provide approval to merge code into the master branch in GitHub

5. Merge code with the master branch

===

Difference b/w fetch and pull

git pull - taking a code directly from the remote repo to the Working directory (git fetch + merge)
git fetch - taking the code first on local repo from remote repo(.git) and after we need to use merge command

==
Remote Repository URL: https://github.com/Ranjeetkendre/GitDemo

UN : Ranjeetvelocity@gmail.com
Password - HSBCWelcome@123

1)Click + and enter the repository name and click create a repository

2)To check created repository-->Click on Symbol(Left Top)

3)To push the code on repository-->Select any particular Repository--> Copy URL of repository(//github.com/Ranjeetkendre/12Feb_C_Morning.git)

4)Create a new project and create a package and class in it

5)Create local Repository:
right click on project--> Team--> Share project-->select checkbox-->select checkbox-->Create repository-->Finish

6)To see Git Repository:
Click search Icon and enter git repository(view) OR enter git repositories and Press ENTER

7)Commit source code from local machine to local repository
right click on project-->Team-->Add to index-->right click on project-->Team-->Commit--->add commit message(on RHS)-->commit

8.Push source code from local repository(git) to remote repository(bitbucket/GitHub)
right click on project-->Team-->remote-->push-->Enter URL--> Enter UN & PWD-->next-->source ref(Select Master Branch)-->master-->add Specification-->finish

9)Clone repository:
Pull URL:https:https://github.com/Ranjeetkendre/Automation_19Feb.git-->Given by Team lead

10)Go to Eclipse-->Click clone a repository(Symbol)-->Enter Pull URL,UN,PSW-->Click Next-->Select Master-->Next-->Finish

11)Move Project from the local repo to the working directory
File-->import-->Git-->Project from git-->Existing local repo-->select project—>Finish

===

Question :

Git and GitHub

1. What is Git? What is the difference between Git and GitHub?

Answer: Git is a version control system that lets you manage and keep track of your source code history.
GitHub is a cloud-based hosting service that lets you manage Git repositories.
If you have open-source projects that use Git, then GitHub is designed to help you better manage them.

What is Git
Git is a version control system tool used to track changes in or source code.
it is a to manage your code & file history while coordinating work remotely on those files with others

What is Github
Github is a hosting service for git repositories.
Git is the tool while GitHub is the service to use git.

==

2. What is the advantage of using GitHub for Selenium?
Answer: GitHub is a cloud-based hosting service that lets you manage Git repositories;
It helps to have a backup code in case of physical failures.

GitHub supports branching, so we can have multiple versions of code.
GitHub supports project cloning, so it helps in the easy distribution of projects across multiple teams and multiple locations
GitHub supports code pull so anyone with access rights can pull code on a local machine. This can also be integrated with Jenkins.
GitHub supports code push so anyone with access rights can check in code in GitHub central repository.

==

3. How to handle git conflicts?
Answer: Git can handle on its own most merges by using its automatic merging features.
There arises a conflict when two separate branches have made edits to the same line in a file,
or when a file has been deleted in one branch but edited in the other. Conflicts are most likely to happen when
working in a team environment.

Identify the files that have caused the conflict.
Make the necessary changes in the files so that conflict does not arise again.
Add these files by the command git add.
Finally commit the changed file using the command git commit

==

4. Explain different Git commands?

Answer: Below are the most common git commands

1. Initialise a repo
Create an empty git repo or re-initialize an existing one
$ git init [repository path]

2. git remote add [variable name] [Remote Server Link]
This command is used to connect your local repository to the remote server.

3. git clone

This command is used to obtain a repository from an existing URL.
git clone [url]
Example: navigate to your repo path where you want to clone and write the below command in cmd
git clone https://github.com/hverma22/Test2

4. git log
git log
This command is used to list the version history for the current branch.
Example:
git log --online

5. git merge

```
git merge [branch name]
```

This command merges the specified branch’s history into the current branch.

7. Git Stage Files

To stage or simply add files, you need to use the git add command. You can stage individual files:

or all files at once:

```
$ git add.
```

8. Git Status

If you want to see what files have been created, modified, or deleted, Git status will show you a report.

```
$ git status
```

9.git commit

```
git commit -m “[commit message]”
```

This command records or snapshots the file permanently in the version history.

Example:

```
git commit -m “First Commit”
```

10. git push - After you have committed your changes, the next is to push to a remote repository.

```
git push [variable name] master
```

This command sends the committed changes of the master branch to your remote repository.

Example:

Push a local branch for the first time:

```
git push origin master
git push origin master --force
```

After that, then you can just use

```
$ git push
```

11. git config

This command sets the author name and email address respectively to be used with your commits.

```
git config –global user.the name “[name]”
git config –global user.email “[email address]”
```

Example:

```
git config user.name "Hitendra Kuamar Verma"
git config user.email "Hitendra@Hitendra-PC"
```

12. git pull

```
git pull [Repository Link]
```

This command fetches and merges changes on the remote server to your working directory.

Example:

```
git pull https://github.com/hverma22/Test2.git
```

===

5. How to Switch Branches in Git

When you create a new branch then Git automatically switches to the new branch.

If you have multiple branches, then you can easily switch between branches with git checkout:

```
$ git checkout master
$ git checkout develop
$ git checkout my_feature
```

You can get the specific previous version as well.

==

6. What is the difference between SVN & GIT?

Answer: Below are the differences -
Git is a distributed VCS; SVN is a non-distributed VCS.
Git uses multiple repositories including a centralised repository and server, as well as some local repositories;
SVN is a centralised version control system.
The content in Git is stored as metadata; SVN stores files of content.
Git branches are easier to work with than SVN branches.
Git does not have the global revision number feature as SVN has.
Git has better content protection than SVN.
Git was developed for the Linux kernel by Linus Torvalds; SVN was developed by CollabNet, Inc.
Git belongs to the 3rd generation of Version Control tools; SVN belongs to the 2nd generation of Version Control tools

==

7. Suppose there are 10 classes & I want to push only 5 classes, how do you do that?

Answer: Normally we commit to git, all files are going to git but in your scenario push only a single file git.
For this, you have to run a specific command to push the only single file to git.

```
$ git commit -m "Message goes here" filename
```

Example to push to single file to git

```
$ git commit -m "Pushing Only Single file to git" config/file1.txt
```

Let’s take a look at how to push one or two or three files to git in a single commit.

```
$ git commit -m "Message goes here" file1 file2 file3
```

For example to push three files to git

```
$ git commit -m "Pushing Only three files to git" config/file1.txt  config/file2.txt config/file3.txt
```

==

8. Mention the various Git repository hosting functions.

Answer:

Github
Gitlab

Bitbucket
SourceForge
GitEnterprise
===

9. What is the difference between git pull and git fetch?

Answer: 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 fetch is also used for the same purpose but it works in a slightly different way.
When you perform a git fetch, it pulls all new commits from the desired branch and stores it in a new branch in your local repository. If you want to reflect these changes in your target branch, git fetch must be followed with a git merge. Your target branch will only be updated after merging the target branch and fetched branch. Just to make it easy for you, remember the equation below:

Git pull = git fetch + git merge

===

10. What is the function of ‘git config’?
Answer: Git uses your username to associate commits with an identity. The git config command can be used to change your Git
configuration, including your username.

Now explain with an example.
Suppose you want to give a username and email id to associate a commit with an identity so that you can know who has made a particular commit. For that I will use:
git config –global user. name “Your Name”: This command will add a username.
git config –global user.email “Your E-mail Address”: This command will add an email id.

===

11. what is the repository in Git
Ans : the repository is like a data structure used by VCS to store metadata for a set of files and directories.
It contains the collection of the files as well as the history of changes made to those files.
A repository in Git is considered your project folder.
A repository has all the project-related data. Distinct projects have distinct repositories.

===

12 What is a conflict in git
conflicts happen when people make different changes to the same line of the same file,
or when one person edits a file and another person deletes the same file

==

13. Could u explain git workflow :