GIT [Source Code Management] [Version Control Tool]

"Git is an open source distributed version control system and source code management (SCM)

system with an insistence to control small and large projects with speed and efficiency."

Git is a distributed version control tool or source code management

used to track the versions of files and directories.

git commit -m "Message" = this will display message

git status = The git status command is used to display the state of the repository and staging area. It allows us to see the tracked, untracked files and changes.

6. git log = It lets you list the project history, filter it, and search for specific changes.

7. git init = creates empty repo/inlization of git repo

8. git add <file_name> =

9. git rm <file_name> = remove file from the repository

10. git restore <file_name> = restore the deleted file

11. git clone <url> = taking code from centrial repo to local system,it works only once

clone specific branch : git clone -b <branch_name> <git_url>

12. sudo yum install git -y = Install git in your EC2 instance

13. bare repoitory = it act as centrial repo or remote repo,

we can only push and pull the changes to this repo. we cannot run any git operations in bare repo. git init --bare

14. non-bare repository = its a local repo where we can modifie,add,commit. we can run all git operations

15. git push = used to push the changes from local work space to remote repo

16. git pull = used to bring the changes from remote repo and merges to the local work space automatically, acts like update the latest code.

17. tag = tag is a name its given to a set of version of files and directories it indicates the milestone of a project. we can remember tag in future. We want to remember the set of the version of code in future we go for tagging.

git tag <tag_name>

18. git checkout <branch_name> = used to switch the branch or tag or previse versions

19. branch = it is used for parallel development of two teams will work on the same piece of code later we can integrate by merging.

20. git merg <file_name> = used to merge two different branchs, merge latest changes.

21. merging confit = it will occur when two same piece of code will changes in two different branches, when we try to merge two branch which one having same piece of code at that time merging confit occurs.

solution = i dont know who has made changes, contact the developers who made the modifications in the code by using "git log". they will discuss which changes will need to keep in the file, then we will keep the changes and continue.

23. git merg = is usd to merg the latest changes with diff branch it will create new commit, it pervious complete history is stored in cronolaogical order

24. git revert = used to undo the changes in previous commit but it does not delete history, we can track who has done the revert coz it will create new commit id with each revert.

git revert commit-id

25. git rebase = is nothing but merge used to integrate changes from one branch to another ,if we use rebase history will be added linearly.

26. git reset = used to reset previse commit, ex :if any new commit ID will contents new file if u r resetting the commit that will delete all the files from git repo, stanging area and from work space, there will be no cloue that ur commit coz history will also get deleted.

git reset <commit_ID>

27. git cherrypick = used to merge or update specific commit.

git cherry-pick <commit-id>

ex: if there 10 committ on branch two if i want to update only one commit to my branch then used git cherry-pick

28. git fetch = it will bring changes from remote repo and stores in a separate branch(fetch head) we can review the changes and merge manually if required

29. git squash = used to combine the number of commits to a single commit.

ex: git reset --soft HEAD~N &&git commit

30. git stash = if im working on one branch, if i get critical work on another branch i need to switch to on another branch without committing in completing work. if i switch to another branch the file on the staging area that will be visible on other branches, too avoid wrong commit on other branch i'll stash it before switching to other branch, so this will store in temporary variable. Once i finish the work on branch I'll swatch back to the private branch which i was working. i can revert back to my work by running command

git stash pop.

31. branches can be created for multiple reasons development will be going on development branch

once the code is ready for the first release

we create release1 from the development branch and we make a 1st release from releas1

similarly, development will be going on development branch,

whatever issue we get on relase1 we fix it on release1 branch

it acts as a maintances branch

once the code is ready for 2nd release on the development branch before we create relase2 branch we will merge relase1 branch to development branch then we create release2 branch from development branch

the issue we faced from relase1 should not be visible on relase2 branch and as well as for future releases.

Build and Release = * - build is untested executable file

* - release is a tested build which is ready to release to a customer

build tools = make : its a build tool used to generate executable from C/C++ (.exe files)

ant/Maven : used to generate executable form jar/war/ear

git hooks = it will import some policy before commit and after commit

pre commit hook : git will execue this file when i run git commit

post commit hook : git post-commit hook is called immediately after the commit-msg hook.

It can't change the outcome of the git commit operation, so it's used primarily for notification purposes.

zombie - when a child process doesn't not return its status to parent then it becomes zombie.

orphan - when parent process dies before child process complete exaction then child process become orphan.

Basic Questions

1. What is the difference between Git and SVN?

Git	SVN
Git is a Decentralized Version Control tool	SVN is a Centralized Version Control tool
It belongs to the 3rd generation of Version Control tools	It belongs to the 2nd generation of Version Control tools
Clients can clone entire repositories on their local systems	Version history is stored on a server-side repository
Commits are possible even if offline	Only online commits are allowed
Push/pull operations are faster	Push/pull operations are slower
Works are shared automatically by commit	Nothing is shared automatically

2. What is Git?

Git is a distributed version control system that allows you to track changes in files and revert to any point in time. It is used to help coordinate the work of multiple people on a project while also tracking progress over time.

3. **What is a distributed VCS?**

In Distributed VCS, every contributor can get a local copy or "clone" of the main repository.

every programmer can maintain a local repository which is actually the copy or clone of the central repository which is present on their hard drive. They can commit and update their local repository without any hassles.

With an operation called "pull", they can update their local repositories with new data from the central server.

4. What is the difference between Git and Github?

Git is a version control system of distributed nature that is used to track changes in source code during software development.

GitHub provides a Web-based graphical interface. It also provides access control and several collaboration features, basic task management tools for every project.

5. What are the benefits of using a Version Control System?

- With the Version Control System(VCS), all the team members are allowed to work freely on any file at any time. VCS gives you the flexibility to merge all the changes into a common version.
- All the previous versions and variants are neatly packed up inside the VCS. You can request any version at any time as per your requirement and you'll have a snapshot of the complete project right at hand.
- Whenever you save a new version of your project, your VCS requires you to provide a short description of the changes that you have made. Additionally, you can see what changes are made in the file's content. This helps you to know what changes have been made in the project and by whom.
- A distributed VCS like Git allows all the team members to have a complete history of the project so if there is a breakdown in the central server you can use any of your teammate's local Git repository.

6. What language is used in Git?

Git uses 'C' language. GIT is fast, and 'C' language makes this possible by reducing the overhead of run times associated with high-level languages.

7. What is the purpose of branching in Git?

The purpose of branching in Git is to allow you to build your own branch and move between them. It will enable you to return to past work while keeping your current work intact.

8. How do you create a Repository in Git?

To create a repository, you need to create a directory for the project if it does not already exist, and then simply execute the command "**git init**". By executing this command, a .git

directory will be created inside the project directory i.e. now your project directory has turned into a Git repository.

9. What is a .git Directory?

This .git directory contains all the metadata of the repository and maintains a track of all the changes made to the files in your repository, by keeping a commit history.

All the information regarding commits, hooks, refs, object databases, remote repository addresses, etc. are kept inside this folder. This is the most crucial part of Git. When you clone any Git repository on your local machine, this .git is the directory that actually gets copied.

What is a commit message and what is the command used to write a commit message?

The command used for passing on a message to a git commit is *git commit -m "commit message".* The flag *m* is used to pass a commit message.

If you need to commit new files for the first time, use '*git add <file>*' before '*git commit –a*'.

10. How can you fix a broken commit?

command is : "git commit --amend".

When you run this command, you can fix the broken commit message in the editor.

11. What is a repository in Git?

Repository in Git is a place where Git stores all the files. Git can store the files either on the local repository or on the remote repository.

12. What is 'bare repository in Git?

- A bare git repository is intended to be used as a remote repository where code is shared between members of the team. A bare repo is nothing but the .get folder

- After you issue a git init --bare command, you won't be able to write code in that repo, which also means you won't have need to clean up Git worktrees or workspaces.

git clone –bare git init bare

13. What is a 'conflict' in git?

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.

14. What do you mean by “git cherry-pick”?

If by mistake, you have committed a change into the wrong branch, you can use the “git cherry-pick” command. This command will allow you to apply commit from one branch to another branch.

git cherry-pick <commit id>

What is the function of the command ‘*git rm*‘?

The ‘*git rm*‘ command is used to remove the file from both staging and the disc

What is the purpose of the ‘*git checkout*‘ command in git?

Switch from one branch to another branch.

Git checkout

What is the function of ‘*git log*‘?

The ‘*git log*‘ is used to find specific commits in your project history

What is the command used to delete a branch?
Deleting a branch Locally

You can delete a git branch on your local machine using the command: **git branch -d <local_branch_name>**

Deleting a branch Remotely

You can delete a git branch remotely using the command: **git push origin --delete <remote_branch_name>**

How can you change any older commit message?
To change older commit the command is **=> git rebase -i**

Git commit –amend -m “new commit message” => **edit the most latest commit**

git rebase = is nothing but merge used to integrate changes from one branch to another ,if we use rebase history will be added linearly

How to resolve and solve merge conflicts?

It is very easy to resolve merge conflicts as Git allows you to go back to the previous state. Just use a “git merge --abort” command, and you will be able to undo the merge and start the task again.

Why do you use Subgit?

The Subgit is a popular tool used for stress-free transferring of SVN to Git and it allows using various Git and sub-version features.

What is git instaweb? How is it used?

A git instaweb is a script that helps to set up a temporary instance of GitWeb on a web server for browsing local repositories.

How to resolve a conflict in Git?

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

In Git how do you revert a commit that has already been pushed and made public?

two ways:

- In a new commit, remove or fix the bad file and push it to the remote repository. Then, use the following command to commit it to the remote repository:

git commit –m "commit message"

- Create a new commit to undo all of the changes made in the previous commit. The following command should be used:

git revert <commit id>

15. How do you edit or fix the last commit message in Git?

Ans. If you forget to add anything in the commit message or committed a typo error, you can rectify it by using –amend flag command.

git commit –amend -m "Sorry I missed an important update"

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

Git pull command pulls all new commits from a specific branch in the central repository and makes the target branch in your local repository up-to-date.

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 central repo 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.

git merg = is usd to merg the latest changes with diff branch it will create new commit, it pervious complete history is stored in cronolaogical order

17. What does the commands ‘*git reset –mixed*’ and ‘*git merge –abort*’ perform?

The command ‘*git reset –mixed*’ is used to undo changes to the working directory and git index.

The ‘*git merge –abort*’ command is used to terminate the merge process and return to the state before the merge

18. What is a ‘HEAD’ in git, and how many HEADs can be created in a repository?

The HEAD is simply a reference to the branch’s most recent commit object. There will always be a default HEAD referred to as ‘master’ or now ‘main’ (as per GitHub), but there is no limit to the number of HEADs available. In other words, it can have as many HEADs as it wants.

19. What is ‘staging area’ or ‘index’ in Git?

The staging area can be described as a preview of your next commit. When you create a git commit, Git takes changes that are in the staging area and make them as a new commit. You are allowed to add and remove changes from the staging area. The staging area can be considered as a real area where git stores the changes.

How will you find out if a branch has already been merged or not?

- **git branch –merged master –** it will list all the branches that have been renamed into master.
- **git branch –merged** – it lists the branches that have been merged into the current branch (HEAD).
- **git branch –no-merged** – it lists the branches that have not been merged.

What work is restored when the deleted branch is recovered?

The files which were stashed and saved in the stash index list will be recovered back. Any untracked files will be lost. Also, it is a good idea to always stage and commit your work or stash them. if you want to fetch the log references of a particular branch or tag then run the command – “git reflog <ref_name>”.

What is the difference between the ‘git diff ’and ‘git status’?

‘git diff’ is similar to ‘git status’, the only difference is that it shows the differences between various commits and also between the working directory and index.

What is the difference between ‘git remote’ and ‘git clone’?

‘git remote add’ creates an entry in your git config that specifies a name for a particular URL whereas ‘git clone’ creates a new git repository by copying an existing one located at the URL.

What is git stash drop?

Git ‘stash drop’ command is used to remove the stashed item. It will remove the last added stash item by default, and it can also remove a specific item if you include it as an argument.

example.

If you want to remove a particular stash item from the list of stashed items you can use the below commands:

git stash list: It will display the list of stashed items like:
stash@{0}: WIP on master: 049d078 added the index file
stash@{1}: WIP on master: c264051 Revert “added file_size”
stash@{2}: WIP on master: 21d80a5 added number to log

If you want to remove an item named stash@{0} use command **git stash drop stash@{0}**.

How do you find a list of files that have changed in a particular commit?

To get a list file that has changed in a particular commit use the below command:

git diff-tree -r *

What is the function of ‘git config’?

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.

EXAMPLE:

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.

Describe the branching strategies you have used.

- branches can be created for multiple reasons development will be going on development branch, once the code is ready for the first release
- we create releas1 from the development branch and we make a 1st release from releas1 branch
- similarly, development will be going on development branch,
- whatever issue we get on relase1 we fix it on release1 branch
- it acts as a maintainers branch
- once the code is ready for 2nd release on the development branch before we create relase2 branch, we will merge relase1 branch to development branch then we create release2 branch from development branch
- the issue we faced from relase1 should not be visible on relase2 and future releases.

In Git, how would you return a commit that has just been pushed and made open?

One or more commits can be reverted through the use of git revert. This command, in a true sense, creates a new commit with patches that cancel out the changes introduced in specific commits. If in case the commit that needs to be reverted has already been published or changing the repository history is not an option then in such cases, git revert can be used to revert commits. If you run the following command then it will revert the last two commits:

git revert HEAD~2..HEAD

What is the meaning of the commands – git status, git log, git diff, git revert <commit>, git reset <file>?

Command	Meaning
git status	Gives a list of which files are staged, unstaged, and untracked
git log	Illustrates the entire commit history by using the default format
git diff	Displays the unstaged changes between index and working directory
git revert <commit>	Undo all the changes made in <commit> and apply it to the current branch by creating a new commit
git reset <file>	Removes <file> from a staging area without overwriting any changes by keeping the working directory unchanged

Name some Git repository hosting functions.

- Github
- Gitlab
- Bitbucket
- GitEnterprise

Why are the Git Stash Drop and Git Stash Clear commands used?

Git Stash drop command or <stash_id> is used to remove a particular stash that is not required.

Git stash clear command is used when all the stashes are to be removed in one go from the repository.

What is the difference between revert and reset?

Revert	Reset
It creates a new commit that undoes the changes made in the previous commit.	It undoes the local changes that have been made to a Git repository.
New history is added to the project and the existing history is not modified.	This command may alter existing history. Reset command operates on the commit his index, and the working directory.
Command: git revert	Command: git reset

What is the difference between Git stash apply and Git stash pop?

The 'Git stash apply' and 'Git stash pop' commands are used when you have to reapply the stashed changes and start working from where you left.

The difference between them is that while the 'Git stash apply' command keeps the changes in the stash list for later use, the 'Git stash pop' command removes the changes from the stash after applying it.

Explain the role of the git annotate command.?

The git annotate command tracks each line of the file based on the commit information. It annotates each line within the given file with information from the commit which introduced that change. The annotate command can also annotate from a given revision .git annotate command:

git annotate [<options>] <file> [<revision>]

How to squash the last N commits into a single commit?

The following are the two ways to squash the last N commits into a single commit:

- Use the below command to write the new commit message from scratch

git reset –soft HEAD~N &&git commit -m

- To start editing the new commit message with a concatenation of the existing commit messages, you will have to get those messages and pass them to commit:

git reset –soft HEAD~N &&git commit –edit -m”$(git log –format=%B –reverse .HEAD@{N})”