Git Interview Q&A

-By SivaReddy

What is Git and its advantages?

- ***DVCS*** *(distributed version control system)*
- ***Branching and Merging*** *– Easy context switching, feature based workflow and disposable experimentation*
- ***Simple and Fast*** *– All the operation performed locally*
- ***Data Consistency*** *- Git ensures the cryptographic integrity of every bit of your project*
- ***Stage Area(Index)*** *- This is an intermediate area where commits can be formatted and reviewed before completing the commit*
- ***Free and Open Source***

What is difference between SVN vs GIT?

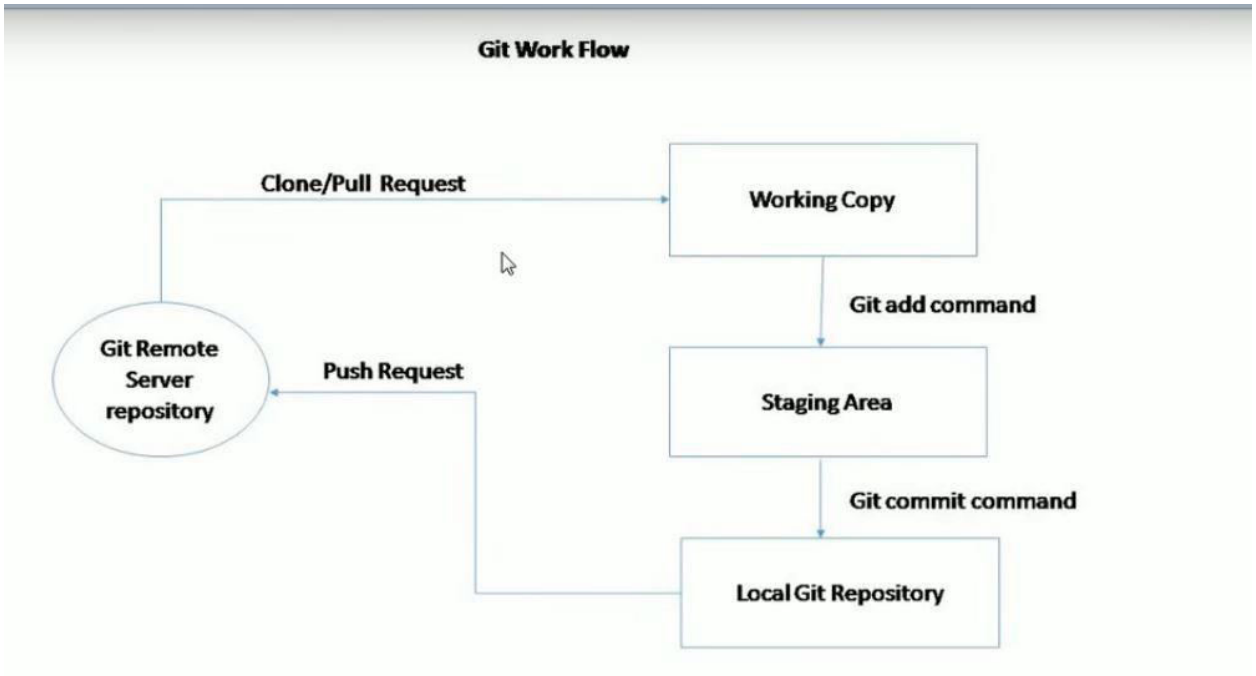
SVN*: Centralizedversion control system*

- *Need an internet connection to commit the files to central repository*
- *Central server goes down, developers cannot continue with their check-ins*

GIT*: Distributed version control system*

- *Supports non-linear development*
- *Local repo is clone of central repository*
- *No need of internet connection to commit as all operations done locally*
- *Branching, Merging easy*

What is git workflow?



- *Create a "repository" for a project with a git hosting tool (like github)*
- *Clone the repository to your local machine – git clone*
- *Add a file to your local repo(workspace) and "commit" (save) the changes – git add, git commit*
- *"Push" your changes to your master branch – git push*
- *Make a change to your file with a git hosting tool and commit*
- *"Pull" the changes to your local machine – git pull*
- *Create a "branch" (version), make a change, commit the change - git branch, git add, git commit*
- *Open a "pull request" (propose changes to the master branch)*
- *"Merge" your branch to the master branch – git merge*

What is clone Command?

git-clone *- Clone a repository into a new directory*

git clone <remote repository URL><Local directory Name>

E.g.: git clone https://github.com/sivajavatechie/JenkinsTest.git **JenkinsClone**

To clone specific branch from the remote repository use git clone -b

git clone -b master https://github.com/sivajavatechie/JenkinsTest.git **Master**

What is difference between clone vs. pull?

git clone *is used to get a local copy of an existing remote repository to work on. It's usually only used once for a given repository, unless you want to have multiple working copies of it around*

git pull *(or git fetch + git merge) is how you update that local copy with new commits from the remote repository. If you are collaborating with others, it is a command that you will run frequently***.**

what is purpose of checkout?

git-checkout - Switch branches or restore working tree files

E. g1: git checkout mybranch – git switch to mybranch

E. g2: git checkout . – It restoes the tree files from remote repository to local repo

E. g 3: git checkout commit_Id -- <filename> - This restores the commit_Id version file from remote to Local repo

what is stash command and why we use it?

git-stash - Stash the changes in a dirty working directory away

E.g.: git stash / push- *The command saves your local modifications away and reverts the working directory to match the HEAD commit.*

E.g. 2: git stash show – *list the modifications stashed away*

E.g. 3: git stash apply – *To restore the modifications stashed away*

what is merge command?

git-merge - Join two or more development histories together

Let us consider branch BugFix is created from master branch

- *Code fix done in BugFix branch*
- *Now you need merge the BugFix branch to master branch*
- *You need to use below command from master branch*

git merge BugFix – *This command merges the code change done in BugFix branch into master branch*

What is difference between branch vs. Tag?

- *branches are symbolic names for line of development. New commits are created on top of branch. The branch pointer naturally advances, pointing to newer and newer commits.*
- *tags are symbolic names for a given revision. They always point to the same object (usually: to the same revision); they do not change.*

what is stage area(Index)?

Stage area holds the files which will be part of the next commit. Git keeps track of the files which were added to the staging area. This help the developer to keep track of the files those will be part of next commit

What is difference betweenlocal and Remote repositories?

- *Git local repository is the one on which we will make local changes, typically this local repository is on our computer.*
- *Git remote repository is the one of the server acts as central repository from where all the developer works(commit/push/pull/clone) for the collaboration*

How to remove files added to staging area?

git reset [option] [commit_id] command is used to alter the staged snapshot and/or the working directory

--soft *-Does not touch the index file or the working tree at all (but resets the head to <commit>, just like all modes do). This leaves all your changed files "Changes to be committed", as git status would put it.*

--mixed *- Resets the index but not the working tree (i.e., the changed files are preserved but not marked for commit) and reports what has not been updated. This is the default action.*

--hard *- Resets the index and working tree. Any changes to tracked files in the working tree since <commit> are discarded.*

How to remove the files committed to the local repository?

git rm <filename>- *This command removes the file from the repo but also deletes it from the local file system*

git rm –cached <file name> - *This command removes the file only from the Git repository and not remove it from the filesystem*

How to check the local modified files in Git?

*Use **git status** command - Displays paths that have differences between the index file and the current HEAD commit, paths that have differences between the working tree and the index file, and paths in the working tree that are not tracked by Git*

How to check the history of the files?

gitk and git log commands

***gitk** - Displays changes in a repository or a selected set of commits. This includes visualizing the commit graph, showing information related to each commit, and the files in the trees of each revision.*

***git log** - It provides a list of all the commits made on our branch with the most recent commit first*

What is difference between Revert/Reset/Checkout?

git revert - *A revert is an operation that takes a specified commit and creates a new commit which inverses the specified commit. git revertcan only be run at a commit level scope and has no file level functionality.*

git reset – *git reset is a simple way to undo changes that haven't been shared with anyone else.*

git checkout - *The git checkout command is used to update the state of the repository to a specific point in the projects history*

Command	***Scope***	***Common use cases***
git reset	*Commit-level*	*Discard commits in a private branch or throw away uncommited changes*
git reset	*File-level*	*Unstage a file*
git checkout	*Commit-level*	*Switch between branches or inspect old snapshots*
git checkout	*File-level*	*Discard changes in the working directory*
git revert	*Commit-level*	*Undo commits in a public branch*
git revert	*File-level*	*(N/A)*

What is difference between merging vs. Rebasing?

git merge and rebase commands are designed to integrate changes from one branch into another branch

git merge *is easy because it's a non-destructive operation. The existing branches are not changed in any way*

git rebase - *The major benefit of rebasing is that you get a much cleaner project history. First, it eliminates the unnecessary merge commits required by git merge and second one is perfectly linear project history*

What is git hook?

Git has a way to fire off custom scripts when certain important actions occur. here are two groups of these hooks:

- *client-side - Client-side hooks are triggered by operations such as committing and merging*
- *server-side - server-side hooks run on network operations such as receiving pushed commits. You can use these hooks for all sorts of reasons.*

All these scripts are stored under the git install sub directory @. git/hooks.

You can write scripts using python/Ruby/shell

Useful git commands

git help -*list available subcommands and someconcept guides.*

git add <filename> - *Add file to staging area/index*

git add . *– add all modified/untracked files to staging area/index*

git commit -m "commit message" – *commit the files to local repo*

git commit -a - *Commit any files you've added with git add, and also commit any files you've changed since then*

git push origin master *– Send changes to the master branch of your remote repository*

git status *- List the files you've changed and those you still need to add or commit*

git checkout -b <branch name> *- It creates new branch*

git checkout <branch name> *- switched to new branch*

git branch *- list all the branches available under the repository*

git clone *– Clone remote Repository into your Local Workspace*

git pull *– pull command performs two operations: it fetches changes from a remote branch, then merges them into the current branch.*

git fetch- git fetch *command without merging them in the current working branch*

git push *– Push your local repo changes to Remote repository*

gitk - *will show a nice graphical representation of the resulting history*

git log - *git log command can list history of the commits*

git show*<commit id>* *- displays the details about the particular commit*

git diff *– compare the current HEAD to local workspace and display any diff found*

git grep "<search string> -*Search for the string in any version of the project*