**Git**

**What is the difference between Git and SVN?**

|  |  |
| --- | --- |
| **Git** | **SVN** |
| distributed VC | Centralized VC |
| Push and pull are fast | Push and pull are slow |
| Can clone entire repo | Version history stored in server |
| Offline commit is possible | Online commit is possible |
| 3rd generation VC | 2 nd generation VC |

**What is Git?**

Git is distributive VC system and source code management .We can track changes of file and even revert back.

**What is the command to write a commit message in Git?**

git commit –a

**What is ‘bare repository’ in Git?**

A “bare” repository in Git just contains the version control information and no working files, It does not contain .git subdirectory. Instead, it contains all the contents of the .git sub-directory directly in the main directory .

But working directory consist of below.

1. .git subdirectory with all the Git related revision history
2. A working tree, or checked out copies of your project files.

**What language is used in Git?**

C language

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

**git commit -m “commit message”  : Commit**

**git revert <name of bad commit> : revert**

**What is the difference between git pull and git fetch?**

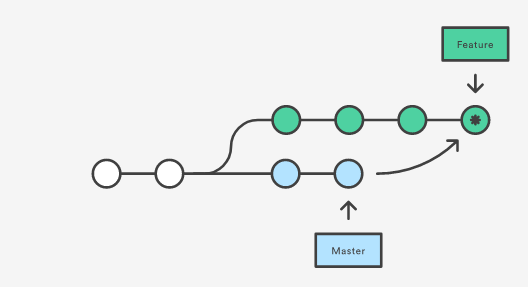
**git Pull : Pulls new changes from particular branch and store in local directory target branch.**

**Git fetch :** it pulls all new commits from the desired branch and stores it in a new branch in your local repository. So to reflect it in target branch need to user

**git merge.**

**Git pull** = **git fetch+ Git merge**

**Merge :** Existing branch not changed.just will connect master to feature.

****

**What is ‘staging area’ or ‘index’ in Git?**

That before completing the commits, it can be formatted and reviewed in an intermediate area known as ‘Staging Area’ or ‘Index’.

**What is Git stash?**

Use git stash when you want to record the current state of the working directory and the index, but want to go back to a clean working directory.

**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.

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 has changed in a particular commit?**

**git diff-tree -r {hashcode}**

This will list all the files that were changed or added in that commit.

**-r flag** :list individual files, rather collapsing them into root directory names.

**git diff-tree –no-commit-id –name-only -r {hash}**

**–no-commit-id :** suppress the commit hashes from appearing in the output.

**–name-only : Gives only name instead of path.**

**What is the function of ‘git config’ ?**

The git config command can be used to change your Git configuration, including your username.

**git config –global user.name “Your Name”:**This command will add username.  
**git config –global user.email “Your E-mail Address”:**This command will add email id.

**What does commit object contains?**

* A set of files.
* Reference to parent commit objects.
* 40 character string that uniquely identifies the commit object.

**How can you create a repository in Git?**

**git init**

**How do you squash last N commits into a single commit?**

**git reset –soft HEAD~N &&**

**git commit**

**What is Git bisect? How can you use it to determine the source of a (regression) bug?**

Bisect is used find the bug in commit. This uses binary search algorithm to find bug.

**git bisect <subcommand> <options>**

**Describe branching strategies you have used?**

**Feature branching :**

A feature branch model keeps all of the changes for a particular feature inside of a branch. When the feature is fully tested and validated then task will merged to master.

**Task branching**: Here each branch is implemented on own branch with task key.

**Release branching:** Once the develop branch has acquired enough features for a release, you can clone that branch to form a Release branch. Creating this branch starts the next release cycle, so no new features can be added after this point, only bug fixes, documentation generation.

**How will you know in Git if a branch has already been merged into master?**

**git branch –merged** It lists the branches that have been merged into the current branch.  
**git branch –no-merged** It lists the branches that have not been merged.

**What is Git rebase and how can it be used to resolve conflicts in a feature branch before merge?**

**git rebase** is a command which will merge another branch into the branch where you are currently working, and move all of the local commits that are ahead of the rebased branch.

If a feature branch was created from the master, and since then the master branch has received new commits, Git rebase can be used to move the feature branch to the tip of master. And move new commit to ahead of the feature commit.

git rebase [new-commit]

**What is Git fork? What is difference between fork and branch? How to create tag?**

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

**How to revert previous commit in git?**

git reset — hard HEAD

Head : last commit in current branch.

**Feature and branch branch ?**

**Feature:All feature/ new functions are done in feature branch.**

A **branch in Git** is simply a lightweight movable pointer to one of these commits. The default **branch**name in **Git** is master

**What is a ‘conflict’ in git?**

**merge conflicts : This occur when 2 person pulled from master and   
one made changes ,pushed to git and another doesn’t know this and tried to push the commit .**

1. **Using git status we can see where conflict occurs.**
2. **Use git mergetool to resolve the issue**

+----------------------+

| | | |

|LOCAL |BASE |REMOTE |Local: file from current branch

| | | |Base: how change looked before Local

+----------------------+ Remote : changes need to merge now

| MERGED |Merged merge result, this is what gets saved in the repo

| |

+----------------------+

**Edit data in merged and exit from Vi.**

1. **Commit the message.**

**Git clone vs Git pull ?**

**Git Clone :** used to get local copy of the repository. Used only once for repo until you want multiple copies.

**Git Pull :** used to update local copy with new commits.

**How can you bring a new feature in the main branch?**

To bring a new feature in the main branch, you can use a command “git merge”

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

**git remote add :** will create an entry in git.

**Git clone :** copy the data from central repo.

**What is the function of ‘git diff ’ in git?**

‘git diff ’ shows the changes between commits, commit and working tree etc.

**Git checkout ?**

Used to switch from brach to branch.

**What is the function of ‘git rm’?**

Used to remove file from staging area.