1. What is git?

Git is a Distributed Version Control System for tracking changes in source code during software development. Git also controls the single point of failure. Git can have multiple repositories created in which all the files of corresponding repository or project can be placed. Git supports branching strategy with which hassle free source code maintenance can be done.

1. What is the difference between git, github and gitlab?

|  |  |  |
| --- | --- | --- |
| **Git** | **Github** | **Gitlab** |
| Git is a distributed version control system which tracks changes to source code over time. | GitHub is a web based hosting service for Git repository to bring teams together. | GitLab is a web based devops lifecycle tool that provides a Git repository manager |
| Git is a command line tool that requires an interface to interact. | GitHub is a graphical interface and a development platform | Provides more convenient UI than GitHub |
| Git creates a local repository to track changes locally rather than to store them on a centralized server. | GitHub is open-source which means code is stored in a centralized server and is accessible to everybody. | GitLab is an open-source end-end software development platform with built-in version control, issue tracking, code review, CI/CD and more. |
| Git can work without GitHub as other web-based Git repositories are also available. | GitHub is the most popular Git server but there are other alternatives available such as GitLab and BitBucket. | GitLab is less popular than that of GitHub. |
| Installed locally | Hosted in the cloud | Can be self-hosted on your own servers, in a container, or on a cloud provider. |

1. Any other version control tools other than git?

Yes.

**CVS : Cenralized Version Control System (Only one master copy of software is used)**

**SVN : Centralized Version Control System (Only one master copy of software is used)**

**GIT** : Distributed Version Control System (Can be cloned and branched)

**Mercurial** : Distributed Version Control System similar to GIT and initially designed as a source for larger development programs.

**Bazaar** : Bazaar is a Distributed Version Control System similar to Git and

Mercurial. Bazaar is unique that it can be deployed either with a

Central code base or as a distributed code base.

1. What is the difference between SVN and git?

SVN is a Centralized Version Control System which uses only one master copy of software wherein changes to the files can be commited by checkout and checkin to the master copy. It can lead to single point of failure.

Git is a Distributed Vesion Control System which can be used to clone the entire project to your local machine and can secure the master copy of the Project by allowing the use of branching strategy. It is hassle free and can nullify the single point of failure.

1. What is merge conflicts have you faced ever in you experience? if you face how you resolve?

A Merge conflict is an event that occurs when Git is unable to automatically resolve differences in code between two commits. Conflict occurs if the commits update the same line of code differently.

* If difference in code between two commits are less(only in few places)

When merge conflict arise then open the file that has merge conflict and correct the code at all places deciding whether to keep others code/our code/both code by looking for the >>>>>>>> ========== <<<<<<<< places.

Once corrections are made conflicts will be resolved as there will be no more differences in code, finally merge the code.

* If difference in code between two commits are more(more places)

Delete the branch which is causing the conflicts. Create a new copy of master, make the corresponding changes and Perform merge.

Perform the rebase inside your feature branch with

git pull origin master –rebase

Resolve the conflicts then perform Merge.

1. What is git stash?

Git stash is used when we want to save the current state of the working directory and the index.

Commands for git stash:

--------------------------------

git statsh – To record the current state of the Working directory

git stash push – To record the current state of the Working directory

git stash pop – To retrive saved files to the working directory

git stash branch <branch\_name> -- To persorm stash on the branch

git stash show – To display the list of files/work that has been stashed

git stash clear – Clears all the stash

git stash drop – To remove the stash entry from the list of stash entries

1. What is git head?

Git head is a reference pointer pointing to the last commit in the current checkout branch.

1. What is staging area in git?

A staging area in git is the place where the code lies after git add and before git commit.

1. What is the difference between git fetch, git pull and git clone?

Git Fetch : git fetch is used to derive the information regarding the commits, files and refs from the remote repository. ie; if we want to see what everybody else is working on.

**git fetch <remote>** : Fetch all of the branches from the repository. This also downloads all of the required commits and files from the other repository.

**git fetch <remote> <branch>** : Same as above, but only fetch the specified branch

**git fetch** --all : A power move which fetches all the registered remotes and their branches.

**git fetch –dry-run** : The –dry-run option will perform a demo run of the command. It will output examples of actions it will take during the fetch but not apply them.

Git Pull : git pull = git fetch + git merge

**git pull –no-commit <remote>** : Similar to default invocation, fetches the remote content but does not create a new merge commit.

**git pull –rebase <remote> :** Same as the previous instead of using git merge to integrate the remote branch with the local one, use git rebase.

**git pull --rebase orgin :** This simply moves your local changes onto the top of what everybody else has already contributed.

**Git Clone** : This is used to copy the complete repository from remote to local.

**git clone <git path> :** Complete repo will be Cloned

**git clone <repo> <directory> :** Cloning to a specific folder. Clone the repository located at <repo> into the folder called <directory> on the local machine.

**git clone –branch <tag> <repo> :** Cloning a specific tag.

**Git clone –branch :** Cloning a specific branch