What is Version control?

- **Ans.** It is a system that records changes to a file or set of files over time so that you can recall specific versions later.

What is Version control?

- This is probably the easiest question you will face in the interview. My suggestion is to first give a definition of Version control. It is a system that records changes to a file or set of files over time so that you can recall specific versions later. Version control systems consist of a central shared repository where teammates can commit changes to a file or set of file. Then you can mention the uses of version control.
- Version control allows you to:
- Revert files back to a previous state.
- Revert the entire project back to a previous state.
- Compare changes over time.
- See who last modified something that might be causing a problem.
- Who introduced an issue and when.

What are the benefits of using version control?

- I will suggest you to include the following advantages of version control:
- With Version Control System (VCS), all the team members are allowed to work freely on any file at any time. VCS will later allow you to merge all the changes into a common version.
- All the past versions and variants are neatly packed up inside the VCS. When you need it, you can request any version at any time and you'll have a snapshot of the complete project right at hand.
- Every time you save a new version of your project, your VCS requires you to provide a short description of what was changed. Additionally, you can see what exactly was changed in the file's content. This allows you to know who has made what change in the project.
- A distributed VCS like Git allows all the team members to have 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.

What are the benefits of using version control?

- Following are the benefits with Version Control Systems (VCS).
- **Collaboration:** Each member of a team gains the ability to work freely on a specific file at any time, and the VCS then merges the changes into a common version.
- **Storing Versions:** VCS allows you to store versions of your project, which is an essential habit for you to review the changes that are made from the beginning.
- **Restoring Previous Versions:** VCS enables you to restore back to the earlier versions of the file (or even the complete project) without any mess.
- **Backup:** VCS also acts like a backup of your project. If any of your team member wipes out the entire project, then you can easily recover it from one of your teammate's repositories.

- **What are the main benefits of GIT ?**
- Distributed System: GIT is a Distributed Version Control System (DVCS). So you can keep your private work in adaptation control yet totally escaped others. You can work disconnected too.
- Flexible Workflow:GIT enables you to make your own work process. You can utilize the procedure that is appropriate for your venture. You can go for brought together or ace slave or some other work process.
- Fast: GIT is quick when contrasted with other form control frameworks.
- Data Integrity: Since GIT utilizes SHA1, information isn't less demanding to degenerate.
- Free: It is free for individual utilize. Such huge numbers of beginners utilize it for their underlying activities. It likewise works exceptionally well with substantial size task.
- Collaboration: GIT is anything but difficult to use for ventures in which joint effort is required. Numerous prevalent open source programming over the globe utilize GIT
-

- **Which VCS tool you are comfortable with?**
- You can just mention the VCS tool that you have worked on like this: “I have worked on Git and one major advantage it has over other VCS tools like SVN is that it is a distributed version control system.”
  Distributed VCS tools do not necessarily rely on a central server to store all the versions of a project’s files. Instead, every developer “clones” a copy of a repository and has the full history of the project on their own hard drive.

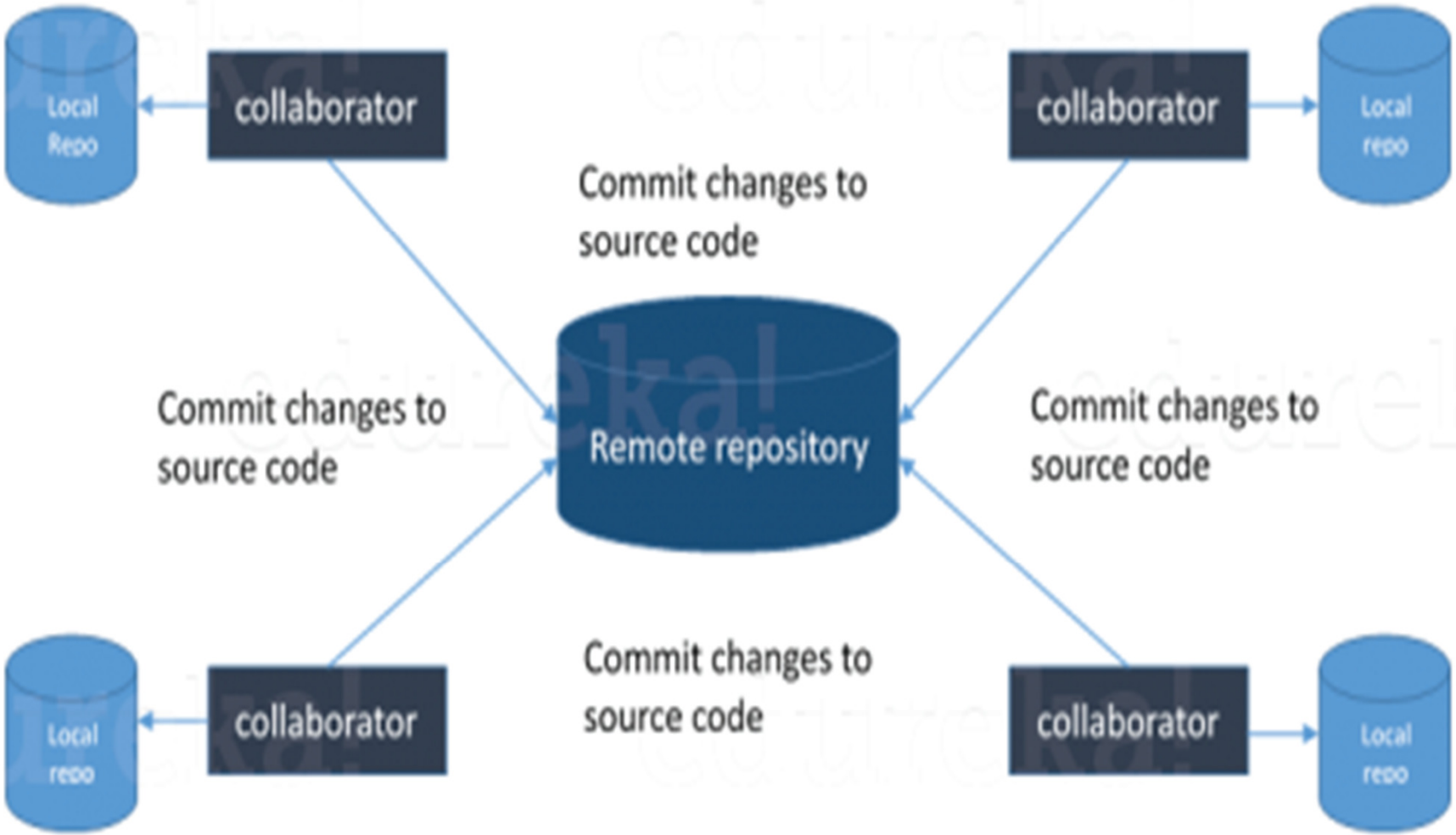
- **What measures we have taken to handle revision (version) control?**
  To handle revision control, post your code on SourceForge or GitHub so everyone can view it and ask the viewers to give suggestions for the better improvement of it.

- **What are the uses of Version control?**
  Revert files back to a previous state.
- Revert the entire project back to a previous state.
- Compare changes over time.
- See who last modified something that might be causing a problem.
- Who introduced an issue and when.

What is Git?

- **Ans.** Git is a Distributed Version Control system (DVCS). It can track changes to a file and allows you to revert back to any particular change

- **What is Git?**
- I will suggest that you attempt this question by first explaining about the architecture of git as shown in the below diagram. You can refer to the explanation given below:
- Git is a Distributed Version Control system (DVCS). It can track changes to a file and allows you to revert back to any particular change.
- Its distributed architecture provides many advantages over other Version Control Systems (VCS) like SVN one major advantage is that it does not rely on a central server to store all the versions of a project's files. Instead, every developer "clones" a copy of a repository I have shown in the diagram below with "Local repository" and has the full history of the project on his hard drive so that when there is a server outage, all you need for recovery is one of your teammate's local Git repository.
- There is a central cloud repository as well where developers can commit changes and share it with other teammates as you can see in the diagram where all collaborators are commiting changes "Remote repository".
- GIT is a distributed version control system and source code management (SCM) system with an emphasis to handle small and large projects with speed and efficiency.


Local Repo
collaborator
Commit changes to source code
collaborator
Local repo
Commit changes to source code
Remote repository
Commit changes to source code
Local repo
collaborator
Commit changes to source code
collaborator
Local repo

- **What is GIT?**
- A) Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

What Are The Advantages Of Using Git?

- a) Data redundancy and replication
  b) High availability
  c) Only one.git directory per repository
  d) Superior disk utilization and network performance
  e) Collaboration friendly
  f) Any sort of projects can use GIT

What are the main advantages of Git over CVS ?

- The biggest advantage is that Git is distributed while CVS is centralised. Changes in CVS are per file, while changes (commits) in Git they always refer to the whole project. Git offers much more tools than CVS.

What is the relationship between GIT and SCM tools?

- A) SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

What is the difference between Git and SVN?

- The proper answer for this according to me will be the architectural differences between Git and SVN. So the basic difference is that Git is distributed and SVN is centralized version control system.
- Then explain the same by including the below mentioned differences:

What is the difference between GIT and SVN?

The difference between GIT and SVN is

- a) Git is less preferred for handling extremely large files or frequently changing binary files while SVN can handle multiple projects stored in the same repository.
- b) GIT does not support ‘commits’ across multiple branches or tags. Subversion allows the creation of folders at any location in the repository layout.
- c) Gits are unchangeable, while Subversion allows committers to treat a tag as a branch and to create multiple revisions under a tag root.

GIT Vs SVN

Git	SVN
In Git we make changes locally, check in (commit) to our local copy of the repository, and then "push" those change sets to another repository when you want to publish them or share them with other users.	When we "check in" or "commit" to SVN, we are contacting a central repository, synchronizing (merging & resolving) with the centralized versions, creating a change list, and then sending that change list back to the centralized repository.
We can "push" changes anywhere even in our teammate's local Git repository. This is how two or more users can collaborate on a new feature without impacting the centralized repository.	All users use and share the same repository, possibly with branching. If two users want to share code, the only way they can do so is by checking into the repository and then each doing a sync.

Why GIT better than Subversion?

- GIT is an open source version control system; it will allow you to run ‘versions’ of a project, which show the changes that were made to the code overtime also it allows you keep the backtrack if necessary and undo those changes.
- Multiple developers can checkout, and upload changes and each change can then be attributed to a specific developer.

Why do you use GIT?

- A) Git is a version control system (VCS) for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development but it can be used to keep track of changes in any set of files.

What is the purpose of Git?

- A) The purpose of Git is to manage a project, or a set of files, as they change over time. Git stores this information in a data structure called a repository.

How will you start GIT for your project ?

- **Answer:** We utilize git init order in a current venture catalog to begin form control for our undertaking. After this, we can utilize git add and git confer orders to add records to our GIT archive.

What language is used in Git?

- Instead of just telling the name of the language, you need to tell the reason for using it as well. I will suggest you to answer this by saying:
- 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.
- Git used “C programing language”.

What is a repository in GIT?

- A repository contains a directory named .git, where git keeps all of its metadata for the repository. The content of the .git directory are private to git.

How can you create a repository in Git?

- In Git, to create a repository, create a directory for the project if it does not exist, and then run command “git init”.
- By running this command .git directory will be created in the project directory, the directory does not need to be empty.

- **How do you rate GIT in terms of speed?**
- A) Git is fast. Speed and performance has been a primary design goal of the Git from the start. With Git, nearly all operations are performed locally, giving it a huge speed advantage on centralized systems that constantly have to communicate with a server somewhere.
- Git was built to work on the Linux kernel, meaning that it has had to effectively handle large repositories from day one. Git is written in C, reducing the overhead of runtimes associated with higher-level languages.

Explain some basic Git commands?

- Below are some basic Git commands:

Command	Function
git config --global user.name "name" git config --global user.email "E-mail"	Configure the author name and email address to be used with your commits
Git init	Create a new local local respository
Git clone /path/to/repository	Create a working copy of a local repository
git clone username@host:/path/to/repository	For a remote server, use
git add <Filename.> git add *	Add one or more file to staging
git commit -m "Commit message"	Commit changes to head
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

What is the function of ‘GIT PUSH’ in GIT?

- ‘GIT PUSH’ updates remote refs along with associated objects.

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

Ans. It can be done in two ways:

- By removing or fixing the bad file in a new commit and pushing it to the remote repository. Once the necessary changes to the file has been made, commit it to the remote repository. Use: **git commit -m "commit message"**
- By creating a new commit that undoes all changes that were made in the bad commit. To do this, use command: **git revert <name of bad commit>**

- **In Git how do you revert a commit that has already been pushed and made public?**
- One or more commits can be reverted through the use of *git revert*. This command, in essence, creates a new commit with patches that cancel out the changes introduced in specific commits. In case the commit that needs to be reverted has already been published or changing the repository history is not an option, *git revert* can be used to revert commits. Running the following command will revert the last two commits:
- git revert HEAD~2..HEAD
- Alternatively, one can always checkout the state of a particular commit from the past, and commit it anew.

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

- I will suggest you to first give a small definition of Git bisect, Git bisect is used to find the commit that introduced a bug by using binary search. Command for Git bisect is
  git bisect <subcommand> <options>

- Now since you have mentioned the command above, explain what this command will do, This command uses a binary search algorithm to find which commit in your project's history introduced a bug. You use it by first telling it a "bad" commit that is known to contain the bug, and a "good" commit that is known to be before the bug was introduced.
- Then Git bisect picks a commit between those two endpoints and asks you whether the selected commit is "good" or "bad". It continues narrowing down the range until it finds the exact commit that introduced the change.

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

- There are two options to squash last N commits into a single commit. Include both of the below mentioned options in your answer:
- If you want to write the new commit message from scratch use the following command
  git reset –soft HEAD~N &&
  git commit
- If you want to start editing the new commit message with a concatenation of the existing commit messages then you need to extract those messages and pass them to Git commit for that I will use
  git reset –soft HEAD~N &&
  git commit –edit -m”$(git log –format=%B –reverse .HEAD@{N})”

- **What is Git rebase and how can it be used to resolve conflicts in a feature branch before merge?**
- According to me, you should start by saying 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 to the top of the history on that branch.
  Now once you have defined Git rebase time for an example to show how it can be used to resolve conflicts in a feature branch before merge, if a feature branch was created from 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.
  The command effectively will replay the changes made in the feature branch at the tip of master, allowing conflicts to be resolved in the process. When done with care, this will allow the feature branch to be merged into master with relative ease and sometimes as a simple fast-forward operation.

What does git pull rebase do?

- A) git pull --rebase is allows you to later squash your commits to a few (or one) commits. If you have merges in your (unpushed) history, it is not so easy to do a git rebase later one.

What does git pull origin master do?

- A) git pull origin master pulls the master branch from the remote called origin into your current branch. It only affects your current branch, not your local master branch.

What is git pull origin?

- A) pull is a fetch and a merge. * `git pull origin master` fetches commits from the master branch of the origin remote (into the local origin/master branch), and then it merges origin/master into the branch you currently have checked out.

- **How do you configure a Git repository to run code sanity checking tools right before making commits, and preventing them if the test fails?**
- I will suggest you to first give a small introduction to sanity checking, A sanity or smoke test determines whether it is possible and reasonable to continue testing.
- Now explain how to achieve this, this can be done with a simple script related to the pre-commit hook of the repository. The pre-commit hook is triggered right before a commit is made, even before you are required to enter a commit message.

- In this script one can run other tools, such as linters and perform sanity checks on the changes being committed into the repository.
  Finally give an example, you can refer the below script:

```
#!/bin/sh
files=$(git diff –cached –name-only –diff-filter=ACM | grep '.go$')
if [ -z files ]; then
exit 0
fi
unfmtd=$(gofmt -l $files)
if [ -z unfmtd ]; then
exit 0
fi
echo "Some .go files are not fmt'd"
exit 1
```

  This script checks to see if any .go file that is about to be committed needs to be passed through the standard Go source code formatting tool gofmt. By exiting with a non-zero status, the script effectively prevents the commit from being applied to the repository.

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

- For this answer instead of just telling the command, explain what exactly this command will do so you can say that, To get a list files that has changed in a particular commit use command **git diff-tree -r {hash}**
  Given the commit hash, this will list all the files that were changed or added in that commit. The -r flag makes the command list individual files, rather than collapsing them into root directory names only.
  You can also include the below mention point although it is totally optional but will help in impressing the interviewer.
  The output will also include some extra information, which can be easily suppressed by including two flags:
  git diff-tree –no-commit-id –name-only -r {hash}
  Here –no-commit-id will suppress the commit hashes from appearing in the output, and –name-only will only print the file names, instead of their paths.

How do you setup a script to run every time a repository receives new commits through push?

- There are three ways to configure a script to run every time a repository receives new commits through push, one needs to define either a pre-receive, update, or a post-receive hook depending on when exactly the script needs to be triggered.
- Pre-receive hook in the destination repository is invoked when commits are pushed to it. Any script bound to this hook will be executed before any references are updated. This is a useful hook to run scripts that help enforce development policies.
- Update hook works in a similar manner to pre-receive hook, and is also triggered before any updates are actually made. However, the update hook is called once for every commit that has been pushed to the destination repository.
- Finally, post-receive hook in the repository is invoked after the updates have been accepted into the destination repository. This is an ideal place to configure simple deployment scripts, invoke some continuous integration systems, dispatch notification emails to repository maintainers, etc.
- Hooks are local to every Git repository and are not versioned. Scripts can either be created within the hooks directory inside the “.git” directory, or they can be created elsewhere and links to those scripts can be placed within the directory.

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

- I will suggest you to include both the below mentioned commands:
  git branch –merged lists the branches that have been merged into the current branch.
  git branch –no-merged lists the branches that have not been merged.

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

- Answer to this is pretty straightforward.
- Command that is used to write a commit message is “**git commit -a**”.

- Now explain about -a flag by saying -a on the command line instructs git to commit the new content of all tracked files that have been modified. Also mention you can use “**git add<file>**” before git commit -a if new files need to be committed for the first time.

- **What does git commit a?**
- A) Basically git commit "records changes to the repository" while git push "updates remote refs along with associated objects". So the first one is used in connection with your local repository, while the latter one is used to interact with a remote repository.

What is ‘bare repository’ in Git?

- You are expected to tell the difference between a “working directory” and “bare repository”.
- A “bare” repository in Git just contains the version control information and no working files (no tree) and it doesn’t contain the special .git sub-directory. Instead, it contains all the contents of the .git sub-directory directly in the main directory itself, where as working directory consist of:
- A .git subdirectory with all the Git related revision history of your repo.
- A working tree, or checked out copies of your project files.

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

- There can be two answers to this question and make sure that you include both because any of the below options can be used depending on the situation:
- Remove or fix the bad file in a new commit and push it to the remote repository. This is the most natural way to fix an error. Once you have made necessary changes to the file, commit it to the remote repository for that I will use **git commit -m "commit message"**
- Create a new commit that undoes all changes that were made in the bad commit.to do this I will use a command **git revert <name of bad commit>**

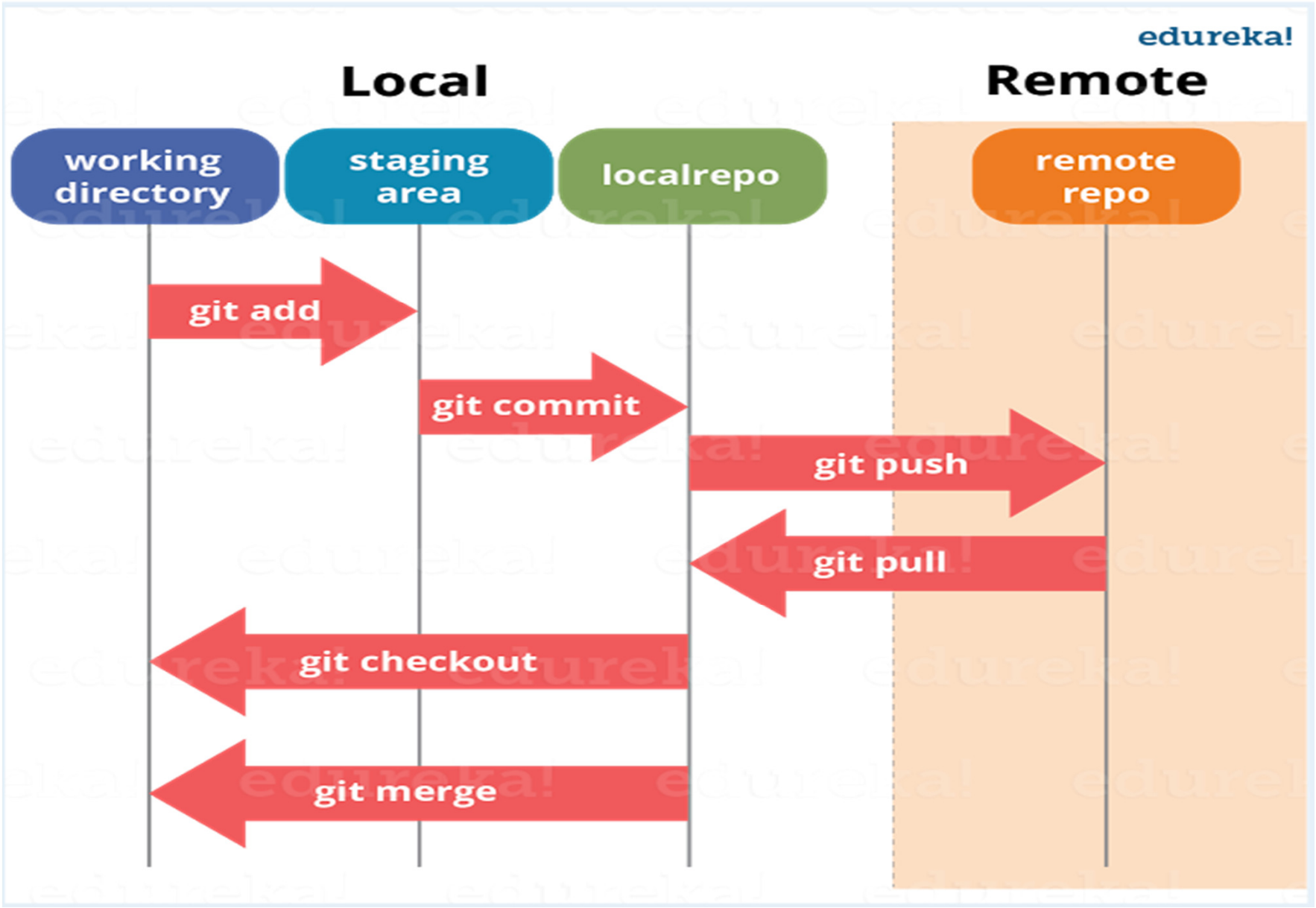
- **What is a pull in git?**
- A) git-pull - Fetch from and integrate with another repository or a local branch
- SYNOPSIS: git pull [options] [[…?]]
- In its default mode, git pull is shorthand for git fetch followed by git merge FETCH_HEAD. More precisely, git pull runs git fetch with the given parameters and calls git merge to merge the retrieved branch heads into the current branch. should be the name of a remote repository as passed to git-fetch.

What is the difference between git pull and git fetch?

- 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 (or)
- Git fetch only downloads new data from a remote repository, but it doesn't integrate any of the downloaded data into your working files. All it does is provide a view of this data.
- git pull downloads as well as merges the data from a remote repository into your local working files. It may also lead to merge conflicts if your local changes are not yet committed. Use the git stash command to hide your local changes.

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

- For this answer try to explain the below diagram as you can see:
- That before completing the commits, it can be formatted and reviewed in an intermediate area known as ‘Staging Area’ or ‘Index’. From the diagram it is evident that every change is first verified in the staging area I have termed it as “stage file” and then that change is committed to the repository.


edureka!
Local
Remote
working directory
staging area
localrepo
remote repo
git add
git commit
git push
git pull
git checkout
git merge

What is Git stash?

- According to me you should first explain the need for Git stash.
- Often, when you've been working on part of your project, things are in a messy state and you want to switch branches for sometime to work on something else. The problem is, you don't want to do a commit of half-done work just so you can get back to this point later. The answer to this issue is Git stash.
- Now explain what is Git stash.
- Stashing takes your working directory that is, your modified tracked files and staged changes and saves it on a stack of unfinished changes that you can reapply at any time.
- GIT stash takes the current state of the working directory and index and puts in on the stack for later and gives you back a clean working directory. So in case if you are in the middle of something and need to jump over to the other job, and at the same time you don't want to lose your current edits then you can use GIT stash.

What is Git stash drop?

- Begin this answer by saying for what purpose we use 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.
- Now give an 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}**.
- When you are done with the stashed item or want to remove it from the list, run the git ‘stash drop’ command. It will remove the last added stash item by default, and it can also remove a specific item if you include as an argument.

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

- For this answer instead of just telling the command, explain what exactly this command will do.
- To get a list files that has changed in a particular commit use the below command:
- **git diff-tree -r {hash}**
- Given the commit hash, this will list all the files that were changed or added in that commit. The -r flag makes the command list individual files, rather than collapsing them into root directory names only.
- You can also include the below mentioned point, although it is totally optional but will help in impressing the interviewer.
- The output will also include some extra information, which can be easily suppressed by including two flags:
- **git diff-tree –no-commit-id –name-only -r {hash}**
- Here –no-commit-id will suppress the commit hashes from appearing in the output, and –name-only will only print the file names, instead of their paths.

What is the function of ‘git config’?

- First tell why we need ‘**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.
- Now explain with an example.
- Suppose you want to give a username and email id to associate 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 username.
  git config –global user.email “Your E-mail Address”: This command will add email id.

What does commit object contains?

- Commit object contains the following components, you should mention all the three points present below:
- A set of files, representing the state of a project at a given point of time
- Reference to parent commit objects
- An SHAI name, a 40 character string that uniquely identifies the commit object.

- **What is Git bisect? How can you use it to determine the source of a (regression) bug?**
- I will suggest you to first give a small definition of Git bisect.
- Git bisect is used to find the commit that introduced a bug by using binary search. Command for Git bisect is
  git bisect <subcommand> <options>

- Now since you have mentioned the command above explain them what this command will do.
- This command uses a binary search algorithm to find which commit in your project's history introduced a bug. You use it by first telling it a "bad" commit that is known to contain the bug, and a "good" commit that is known to be before the bug was introduced. Then Git bisect picks a commit between those two endpoints and asks you whether the selected commit is "good" or "bad". It continues narrowing down the range until it finds the exact commit that introduced the change.

- **Describe branching strategies you have used?**

- This question is asked to test your branching experience with Git so, tell them about how you have used branching in your previous job and what purpose does it serves, you can refer the below mention points:
- 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 by automated tests, the branch is then merged into master.
- Task branching
  In this model each task is implemented on its own branch with the task key included in the branch name. It is easy to see which code implements which task, just look for the task key in the branch name.
- 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, and other release-oriented tasks should go in this branch. Once it is ready to ship, the release gets merged into master and tagged with a version number. In addition, it should be merged back into develop branch, which may have progressed since the release was initiated.

- In the end tell them that branching strategies varies from one organization to another so I know basic branching operations like delete, merge, checking out a branch etc..

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

- The answer is pretty direct.
- To know if a branch has been merged into master or not you can use the below commands:
- **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?**
- According to me you should start by saying 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 to the top of the history on that branch.
- Now, once you have defined Git rebase time for an example to show how it can be used to resolve conflicts in a feature branch before merge.
- 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. The command effectively will replay the changes made in the feature branch at the tip of master, allowing conflicts to be resolved in the process. When done with care, this will allow the feature branch to be merged into master with relative ease and sometimes as a simple fast-forward operation.
-

- **What is SubGit?**
- Begin this answer by explaining what is SubGit used for.
- SubGit is a tool for SVN to Git migration. It creates a writable Git mirror of a local or remote Subversion repository and uses both Subversion and Git as long as you like.
- Now you can include some advantages like you can do a fast one-time import from Subversion to Git or use SubGit within Atlassian Bitbucket Server.We can use SubGit to create a bi-directional Git-SVN mirror of existing Subversion repository. You can push to Git or commit to Subversion at your convenience. Synchronization will be done by SubGit.

- **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.
- A fork is really a Github (not Git) construct to store a clone of the repo in your user account. As a clone, it will contain all the branches in the main repo at the time you made the fork.
- *Create Tag:*
- Click the releases link on our repository page.
- Click on Create a new release or Draft a new release.
- Fill out the form fields, then click Publish release at the bottom.
- After you create your tag on GitHub, you might want to fetch it into your local repository too: git fetch.

What is git cherry-pick?

- Cherry picking in git means to choose a commit from one branch and apply it onto another.
- This is in contrast with other ways such as merge and rebase which normally applies many commits onto a another branch.
- Make sure you are on the branch you want apply the commit to. git checkout master Execute the following:
- git cherry-pick <commit-hash>

How to revert previous commit in git?

- To revert to a previous commit, ignoring any changes:
- git reset — hard HEADwhere HEAD is the last commit in your current branch

How to rebase master in git? Difference between rebase and merge. How to squash or fixup commits?

- Rebasing is the process of moving a branch to a new base commit.
- The golden rule of git rebase is to never use it on public branches. … The only way to synchronize the two master branches is to merge them back together, resulting in an extra merge commit and two sets of commits that contain the same changes.

Explain git stash, pop.

- 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.
- git stash pop applies the top stashed element and removes it from the stack. git stash apply does the same, but leaves it in the stash stack.

- **Branching strategies Pros and Cons — Feature, Release branching, Git flow, Lean Git flow**
- Pro: By keeping latest deployed version in trunk, small fixes can be rolled out quickly without extensive testing of the latest development version.
- Pro: Developers can work more freely in tighter iterations without stepping on eachother's feet.
- Pro: if you have many branches you'll be pushed to adopt a modern DVCS (my experience is with Mercurial but I hear git or Bazaar are also good) rather than stay with a traditional centralized system (like, say, svn).
- Pro: Branches can be used to facilitate 'what-if' scenario's in trying out new code. At the end a decision can be made to merge the new feature or to abandon it.
- Con: Having too many branches in the air at the same time and you start forgetting where things where commited, where changes have been made etc.
- Con: someone has to manage the branch(es) and keep on top of things. In most teams this falls by the way-side.

What is ‘head’ in git and how many heads can be created in a repository?

- A ‘head’ is simply a reference to a commit object. In every repository, there is a default head referred as “Master”. A repository can contain any number of heads.

What are Git repository hosting services you used?

- Github, Bitbucket, Gitlab or you can specify if you have used something else here.

- **What is the function of ‘git reset’?**
- ‘Git Reset’ is to reset your index as well as the working directory to the state of your last commit.

What is the syntax for “Rebasing” in Git?

- Syntax: “git rebase [new-commit] “

What is a ‘conflict’ in git? How to resolve?

- A ‘conflict’ arises when the commit that has to be merged has some change in one place, and the current commit also has a change at the same place. (or)
- A conflict arises when more than one commit that has to be merged has some change in the same place or same line of code. Git will not be able to predict which change should take precedence. This is a git conflict.
- To resolve the conflict in git, edit the files to fix the conflicting changes and then add the resolved files by running git add. After that, to commit the repaired merge, run git commit. Git remembers that you are in the middle of a merge, so it sets the parents of the commit correctly.

What is the command you can use to write a commit message?

- The command that is used to write a commit message is “git commit –a”. The –a on the command line instructs git to commit the new content of all tracked files that have been modified. You can use “git add<file>” before git commit –a if new files need to be committed for the first time.

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

- Git branch—merged lists the branches that have been merged into the current branch
- Git branch—-no merged lists the branches that have not been merged

What is the function of git clone?

- The git clone command creates a copy of an existing Git repository. To get the copy of a central repository, ‘cloning’ is the most common way used by programmers.

What is the function of ‘git config’?

- The ‘git config’ command is a convenient way to set configuration options for your Git installation. Behaviour of a repository, user info, preferences etc. can be defined through this command.

What does commit object contain?

- a) A set of files, representing the state of a project at a given point of time
- b) Reference to parent commit objects
- c) An SHAI name, a 40 character string that uniquely identifies the commit object.

What is ‘head’ in git and how many heads can be created in a repository?

- A ‘head’ is simply a reference to a commit object. In every repository, there is a default head referred as “Master”.
- A repository can contain any number of heads.

What is the purpose of branching in GIT?

- The purpose of branching in GIT is that you can create your own branch and jump between those branches. It will allow you to go to your previous work keeping your recent work intact.

What is the common branching pattern in GIT?

- The common way of creating branch in GIT is to maintain one as “Main“
- branch and create another branch to implement new features. This pattern is particularly useful when there are multiple developers working on a single project.

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” or “git pull command”.

What is a ‘conflict’ in git?

- A ‘conflict’ arises when the commit that has to be merged has some change in one place, and the current commit also has a change at the same place. Git will not be able to predict which change should take precedence.

How can conflict in git resolved?

- To resolve the conflict in git, edit the files to fix the conflicting changes and then add the resolved files by running “git add” after that to commit the repaired merge, run “git commit”. Git remembers that you are in the middle of a merger, so it sets the parents of the commit correctly.

To delete a branch what is the command that is used?

- Once your development branch is merged into the main branch, you don’t need
- development branch. To delete a branch use, the command “git branch –d [head]”.

What is another option for merging in git?

- “Rebasing” is an alternative to merging in git.

What is the syntax for “Rebasing” in Git?

- The syntax used for rebase is “git rebase [new-commit] “

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

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

- **Can you explain about Branching and Merging in GIT?**
- A) The Git feature that really makes it stand apart from nearly every other SCM out there is its branching model.
- Git allows and encourages you to have multiple local branches that can be entirely independent of each other. The creation, merging, and deletion of those lines of development takes seconds.

What is GIT version control?

- With the help of GIT version control, you can track the history of a collection of files and includes the functionality to revert the collection of files to another version. Each version captures a snapshot of the file system at a certain point of time. A collection of files and their complete history are stored in a repository.

Mention some of the best graphical GIT client for LINUX?

- Some of the best GIT client for LINUX is
- a) Git Cola
- b) Git-g
- c) Smart git
- d) Giggle
- e) Git GUI
- f) qGit

What is Subgit? Why to use Subgit?

- ‘Subgit’ is a tool for a smooth, stress-free SVN to Git migration. Subgit is a solution for a company - wide migration from SVN to Git that is:
- a) It is much better than git-svn
- b) No requirement to change the infrastructure that is already placed
- c) Allows to use all git and all sub-version features
- d) Provides genuine stress –free migration experience.

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

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

What is ‘git status’ is used for?

- As ‘Git Status’ shows you the difference between the working directory and the index, it is helpful in understanding a git more comprehensively.

- **What is the difference between the ‘git diff ’and ‘git status’?**
- ‘git diff’ is similar to ‘git status’, but it shows the differences between various commits and also between the working directory and index.

What is the function of ‘git checkout’ in git?

- A ‘git checkout’ command is used to update directories or specific files in your working tree with those from another branch without merging it in the whole branch.

What is the function of ‘git rm’?

- To remove the file from the staging area and also off your disk ‘git rm’ is used.

What is the function of ‘git stash apply’?

- When you want to continue working where you have left your work, ‘git stash apply’ command is used to bring back the saved changes onto the working directory.

What is the use of ‘git log’?

- To find specific commits in your project history- by author, date, content or history ‘git log’ is used.

What is ‘git add’ is used for?

- ‘git add’ adds file changes in your existing directory to your index.

What is the function of ‘git reset’?

- The function of ‘Git Reset’ is to reset your index as well as the working directory to the state of your last commit.

What is git ls-tree?

- ‘git ls-tree’ represents a tree object including the mode and the name of each item and the SHA-1 value of the blob or the tree.

How git instaweb is used?

- ‘Git Instaweb’ automatically directs a web browser and runs webserver with an interface into your local repository.

What does ‘hooks’ consist of in git?

- This directory consists of Shell scripts which are activated after running the corresponding Git commands. For example, git will try to execute the post-commit script after you run a commit.

Explain what is commit message?

- Commit message is a feature of git which appears when you commit a change. Git provides you a text editor where you can enter the modifications made in commits.

How can you fix a broken commit?

- To fix any broken commit, you will use the command “git commit—amend”. By running this command, you can fix the broken commit message in the editor.

Why is it advisable to create an additional commit rather than amending an existing commit?

- There are couple of reason
- a) The amend operation will destroy the state that was previously saved in a commit. If it's just the commit message being changed then that's not an issue. But if the contents are being amended then chances of eliminating something important remains more.
- b) Abusing "git commit- amend" can cause a small commit to grow and acquire unrelated changes.

Name a few Git repository hosting services

- Pikacode
- Visual Studio Online
- GitHub
- GitEnterprise
- SourceForge.net

- One of your teammates accidentally deleted a branch, and has already pushed the changes to the central git repo. There are no other git repos, and none of your other teammates had a local copy. How would you recover this branch?
- Check out the latest commit to this branch in the reflog, and then check it out as a new branch.

- **How can you copy a commit made in one branch to another (e.g. a hot fix commit from released branch to current development branch)?**
- You need to use the cherry-pick command. It provides the possibility to play back an existing commit to your current location/branch. So you need to switch to the target branch (e.g. git checkout development) and callgit cherry-pick {hash of that commit}.
- In spite of applying the same changes, it will be a new commit with a new hash because the changes are applied to a different destination.

- **How do you cherry-pick a merge commit?**
- Cherry-pick uses a diff to find the difference between branches.
- As a merge commit belongs to a different branch, it has two parents and two changesets.
- For example, if you have merge commt ref 63ad84c, you have to specify -m and use parent 1 as a base:
- git checkout release_branch git cherry-pick -m 1 63ad84c

What is the difference between Git and Github?

- A) Git is a revision control system, a tool to manage your source code history.
- GitHub is a hosting service for Git repositories.
- GitHub is a website where you can upload a copy of your Git repository. It is a Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

What are the disadvantages of GIT ?

- **Answer:** GIT has not very many weaknesses. These are the situations when GIT is hard to utilize. Some of these are:
- **Binary Files:** If we have a considerable measure double records (non-content) in our venture, at that point GIT turns out to be moderate. E.g. Tasks with a lot of pictures or Word records.
- **Steep Learning Curve:** It sets aside some time for a newcomer to learn GIT. A portion of the GIT summons is non-instinctive to a fresher.
- **Slow remote speed:** Sometimes the utilization of remote stores in ease back because of system dormancy. Still, GIT is superior to different VCS in speed. Top 50 DevOps Engineer Interview Questions and Answers

What is stored inside a commit object in GIT ?

- **Answer:** GIT confer question contains following data: SHA1 name: A 40 character string to recognize a submit Files: List of documents that speak to the condition of a task at a particular purpose of time Reference: Any reference to parent submit objects.

Why do we create branches in GIT ?

- **Answer:** If we are all the while chipping away at various errands, tasks, deformities or highlights, we require numerous branches. In GIT we can make a different branch for each different reason.
- Let say we are dealing with an element, we make a component branch for that. In the middle of we get a deformity to take a shot at then, we make another branch for imperfection and work on it. Once the deformity work is done, we combine that branch and return to chip away at include branch once more.
- So taking a shot at numerous errands is the fundamental purpose of utilizing various branches.

- **How to delete a Git branch both locally and remotely?**

  To remove a local branch from your local system.

  git branch -d the_local_branch

  To remove a remote branch from the server.

  git push origin :the_remote_branch

- **How do you undo the last commit?**
  git revert commit-id

- **How to Edit an incorrect commit message in Git?**
  git commit --amend -m "This is your new git message"

- **How do you rename the local branch?**
  git branch -m oldBranchName newBranchName

- **How do I remove local files (Not in Repo) from my current Git branch?**

  git clean -f -n

- **How to Checkout remote Git branch?**
  git checkout test

- **How do I remove a Git submodule?**
  git rm the_submodule rm -rf .git/modules/the_submodule

- **How do you create a remote Git branch?**
  git checkout -b your_branch_name git push -u origin your_branch_name

- **How to Change the URL for a remote Git repository?**

  git remote set-url origin git://this.is.new.url

- **How to Change the author of a commit in Git?**

```
git filter-branch -f --env-filter
"GIT_AUTHOR_NAME='NewAuthorName';
GIT_AUTHOR_EMAIL='authoremail@gmail.co
m';
GIT_COMMITTER_NAME='CommiterName';
GIT_COMMITTER_EMAIL='committergmail@g
mail.com';" HEAD
```

- **What is .gitignore?**

  .gitignore tells git which files/folder should be ignore.

  Create a file with name of .gitignore in root.

  temporay files, system files, hidden files or local downloaded modules we can add in .gitignore file, so that git will not added this.

- **How will you explain GitHub?**
- Answer:
  GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside millions of other developers. GIT is famous for its speed and ability to deal with even quite large development projects.

- **What are the features of GitHub?**

- Answer:
  Below is the list of features of GitHub:
  •Creating a folder via the Web Interface: While many of us may manage GitHub repositories through the free GitHub app, GitHub has also built what they called Web Flow. It allows us to manage repositories through GitHub's web interface.
  •Drag and Drop Gist Code: Gist is GitHub's very own facility that allows you to host code snippets.
  •Using GitHub Command Line Interface: GitHub CLI is initiated with a hub. It brings extra commands that can be used along with the GIT commands.
  •Using GIT URL Shortened: To share your GitHub repository when URL is too long.
  •File Finder: Besides creating new files, you can also navigate through the files in any repository quickly.
  •Using GitHub Emoji: Emoji's or emoticons are tiny icons that depict an expression of some sort.
  •Linking Lines: To share and point out specific lines within the file of your repository.
  •Task Checklist: GitHub extends markdown to cater to its own need.
  •Map, CSV, and 3D Rendering: GitHub supports CSV. If you include a .csv file, GitHub will render your CSV file into an interactive tabular data format. It even allows you to search through it. Aside CSV, GitHub will also automatically render Map with the geoJSONformat and 3D with the STL extension.
  •Get Octodex: Octodex is a collection of a creative alternate version of Octocat.

- **How to use GitHub?**
- Answer:

  A user can use GitHub by following ways:
  - Install GIT and create a GitHub account
  - Create a local GIT repository
  - Add a new file to the repository
  - Add a file to the staging environment
  - Create a commit
  - Create a new branch

- **What Is GitHub Link?**
- Answer:
  GitHub Link is version control repository and it also a web-based providing hosting service over the internet.
  GitHub also offers:
  - Distributed Version Control
  - Source Code Management

- **Explain GitHub Workflow?**
- Answer:
  GIT provides three key areas that are uniquely designed, to give developers lots of control over workflow:
  1.Working directory: It contains all the current states of files. Numerous developers can access directory when they are logged in, so collaboration is extremely easy.
  2.Staging Area: It indexes everything for the next commit and any files that have been added or edited since the previous save.
  3.GIT repository is a dedicated space where new commits are added: GIT repository maintains all the metadata, the files, and a dedicated database that tracks versions of the project.

- **More Questions here:**

- https://www.git-tower.com/learn/git/faq

- https://www.onlinefreecourse.net/git-interview-questions-preparation-course-udemy-free-download/