QUIZ-8 GIT- HIMABINDU MELACHURU

1)What is Git ?

Git is widely used version control system for software development. It is a free and open source version control system. Git is speedy with simple design and provides strong support for non-linear development. It identifies its data more like a stream snapshots.

2) In what form is Git being used in your Company ?

Github

It is used as version Control system.

To maintain the code in the central repository and integrate with the automation tools to automate build and deployment.

Different companies use git in different forms, some use, GitHub, some use pure Git on a central server with GitWeb to view repositories in browser, some companies use GitLab

3) What are the different Tools that you integrated Git with in your company ?

Jenkins,Maven

4) What are Git hooks ?

Git hooks are event based triggers. Git hooks are scripts that Git executes before or after events such as: commit, push, and receive. Git hooks are a built-in feature and are run locally. Some of the examples are:

pre-commit: Check the commit message for spelling errors.

pre-receive: Enforce project coding standards.

post-commit: Email/SMS team members of a new commit.

post-receive: Push the code to production.

There are two types of Git hooks client-side and server-side. Client-side hooks are triggered by committing and merging, server-side hooks run on network operations like receiving pushed commits.

5) Give us an example of a Git Hook that you implemented?

enforcing Developers to include Jira ticket number in their commit message, else, commit will fail. This is example of a pre-commit hook.

6) List all commands a developer uses to push to git starting from clean workspace ?

git status

git add <filename>

git commit <filename>

git push origin master

create a repository.

git clone url

git add to add the fles

git commit -m" "

git push

git push

7)Difference between Git push and Git pull

Git push: Transmits the local branch commits to the remote repository branch.

Git pull: Fetch and merge any commits from the tracking remote branch.

8) Different between Git Rebase and git pull

Git rebase: It applies any commits of current branch ahead of specified one. Anything you’ve changed by committing to your current branch but are not in the upstream are saved to a temporary area, so your branch is the same as it was before you started your changes, IE, clean. If you do ‘git pull –rebase’, git will pull down the remote changes, rewind your local branch, then replays all your changes over the top of your current branch one by one, until you’re all up to date. But in git pull, it fetches and merge any commitments from the remote branch.

9) How do you what branch you are currently on ?

git branch : It lists all the branches in the repository but the current branch has an asterix (\*) symbol.

10) How do you what other git branches exist on the git server ?

git branch -a : Lists both local and remote branches

git branch -r: Lists only remote branches.

11) what is a git remote ?

git remote : Lists what branches are on remote servers and managed set of tracked repositories.

It is a management tool to record the remote repositories. It allows you to save long urls as short handles.so that we should not use long urls all the time. It is also used to add, change and delete them

12) How can you temporarily save changes before pulling/merging or switching branches?

git stash : Temporarily save the modified and staged changes.

13) difference between git branches and git tags ?

A git tag represents a version of a particular branch at a moment in time. In git, we don't tag branches but we tag commits in the branches. A git branch represents a separate thread of development that may run concurrently with other development efforts on the same code base.

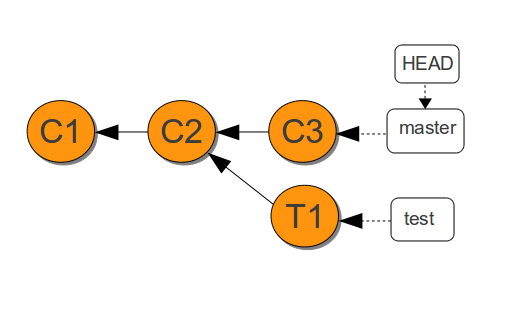
14) what are different types of git tags available ? give exact commands on how to create each type of tag ?

Git uses two main types of tags: lightweight and annotated.

A lightweight tag is very much like a branch that doesn’t change – it’s just a pointer to a specific commit. Annotated tags, however, are stored as full objects in the Git database. They’re check summed and contain the tagger name, email, and date; have a tagging message; and can be signed and verified with GNU Privacy Guard (GPG). It’s generally recommended that you create annotated tags so you can have all this information; but if you want a temporary tag or for some reason don’t want to keep the other information, lightweight tags are available too.

15) what command do you use to merge two branches ? give example

Git merge.

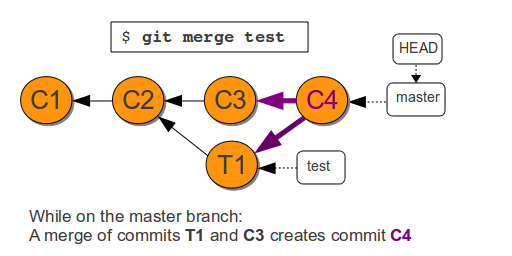


In the diagram, there is a test branch which is deviated from the master branch. The test branch is added and commited to C2. To merge the test branch to the master branch, we use the command git merge.

First to merge any branches, we need to be in the master branch and then give the command to merge.

git checkout master

git merge test



Then the merge of commits T1 and C3 creates another commit called C4 in the master branch. At this point the test branch is merged with the master branch.

16) How do you know current state of your workspace to that on git server ?

git status

17) What command is used to fetch the latest updates by others?

git fetch

git pull

18) What command is used to create a git repo for your personal development?

git init

mkdir

19) Explain the relationship between the working directory, the index, the repository

A working directory is the local space on the hard drive that a repository is mapped to. The working directory generally mirrors the repository structure. A working directory must be set before you can work with files. The git “index” is where you place files you want committed to the git repository. Before you “commit” files to the git repository, you need to first place the files in the git “index”. Git stores this information in a data structure called a repository. A gitrepository contains a set of commit objects.

20) How can a Git merge fail? What do you do?

Git can fail to start the merge and this occurs because git knows there are changes in either your working directory or staging area that could be written over by the files that you are merging in. If this happens, there are no merge conflicts in individual files. You need to modify or stash the files it lists and then try to do a git pull again. Git can fail during merge and this occurs because you have committed changes that are in conflict with someone else's committed changes. Git will do its best to merge the files and will leave things for you to resolve manually in the files it lists. For this problem, we have git merge tools to resolve.

git diff : Shows the difference between two conflicts.

21) what command do you use to know list of previous commits ?

git log

22) What is 3a525393f6a5c47fa08d91ef16c16927ed3cd33a? What are the benefits of this?

All the information needed to represent the history of a project is stored in files referenced by a 40-digit "object name" called SHA 1 hash. It is a unique ID. The advantages are:

* Git can quickly determine whether two objects are identical or not, just by comparing names
* Since object names are computed the same way in every repository, the same content stored in two repositories will always be stored under the same name
* Git can detect errors when it reads an object, by checking that the object's name is still the SHA1 hash of its contents.

23) command to rename a file in Git?

git mv <existing name> <new name>

24) What git command is used to undo changes made to your local repo? What are the variations and how do they work?

git revert <commit>

The git revert command undoes a committed snapshot. But, instead of removing the commit from the project history, it figures out how to undo the changes introduced by the commit and appends a new commit with the resulting content. This prevents Git from losing history, which is important for the integrity of your revision history and for reliable collaboration.

git reset <filename>

If git revert is a safe way to undo changes. When you undo with git reset (and the commits are no longer referenced by any ref or the reflog), there is no way to retrieve the original copy—it is a permanent undo. Care must be taken when using this tool, as it’s one of the only Git commands that has the potential to lose your work.