Git Quiz

Harsh kumar raj

Q1 - What is a version control system?
Sol: Also known as source control or revision control, it is an important software development technique for tracking and managing changes made to code and other files. It is closely related to source control.

There are 2 types of version control system:

1. Distributed VCS
2. Centralized VCS

Q2 - Why did a version control system develop? What were the necessities?
Sol: The idea of being able to deal with a whole lot of files at once took a long time to catch on, as did having more than one person working on a file at the same time.
This was the major concern for developing Version Control System.

Q3 - Define the different types of version control systems.
Sol: 1. Local Version Control System: There is no remote repository. Manage and version all files in your local system only. In this scenario there are no remote servers. All changes are recorded in the local database. As you can see in the image below, each developer has their own computer and they do not share anything.

2. Centralized Version Control System: Here there is a central repository shared by all developers, everyone gets their own working copy. With each commit, your changes are reflected directly in the repository. Unlike distributed systems, developers commit directly to remotes. This means that you may intentionally or unknowingly affect your files.

Q4 - List a few differences between the two version control system types.
Sol:

Centralized Version Control	**Distributed version Control**
Centralized version control is the simplest form of version control in which the central repository of the server provides the latest code to the client machines.	Distributed version control is a form of version control where the complete codebase is mirrored on every developer's computer.
There are no local repositories.	There are local repositories.
Works comparatively slower.	Works faster.

Consider the entire columns for compression.	Consider columns as well as partial columns.
Focuses on synchronizing, tracking, and backing up files.	Focuses on sharing changes.
A failure in the central server terminates all the versions.	A failure in the main server does not affect the development.

Q5 - What is Git?
Sol: Git is a DevOps tool used for source code management. It's a free, open-source version control system that can be used to efficiently handle small to very large projects.

Git is used to track source code changes and allows multiple developers to collaborate on non-linear development.

Q6 - List a few features of Git.

Sol: Tracks history, Free and open source, supports non-linear development, Creates backups, Scalable, Supports collaboration, Branching is easier, Distributed development

Q7 - State any three commands of Git and why we use them.

Sol:

1. **git init:** The git init command is used to create an empty Git repository.

⇨ The git init command creates a .git folder in a directory with several subdirectories. Once the repository is initialized, the process of creating other files begins.

2. git add

⇨ The add command is used to check the status of files and then add them to the staging area.
⇨ Use git add to add new or changed files before running the commit command.

3. git commit

⇨ The commit command ensures that your changes are saved to your local repository.
⇨ git commit –m command will explain each one and help you understand what happened.

Q8 - Is Git the same as Github? Why or Why not?
Sol: No Git and GitHUB is not same.

Here is the reason why?

- ⇨ Git is a version control system that lets you manage and keep track of your source code history. GitHub is a cloud-based hosting service that lets you manage Git repositories.

Q9 - What is the command to get the installed version of Git?
Sol:

- ⇨ [git --version] is used to check the installed version of git.

Q10 - What is the command to add all files and changes of the current folder to the staging environment of the Git repository?
Sol:

- ⇨ [git add .] is used to get all the changes to staging area.

Q11 - What is the difference between git status and git log commands?

Sol:

1. **Git Status** is a useful command for gathering information about the branch you are currently working on.

This command allows you to collect information such as:

- ⇨ Whether the current branch is up-to-date.
- ⇨ All files that are served, not served, or tracked.
- ⇨ All files created, modified, or deleted.
- ⇨ If you have work to do Commit, pull or push

2. The **git log** shows information about your commit history. git log shows the commit history of a branch.

Q12 - What is the command to initialize Git on the current repository?
Sol: **git init** will initialize the Git configuration to the current repository.

Q13 - What are the different states of a file in Git? Explain them along with the associated commands.

Sol: Each file in the working directory can be in one of two states:

Tracked: A tracked file is a file that was included in the last snapshot. These are files known to Git. Each track file can be in one of three substates: **modified, staged, or committed**.

Untracked: Untracked files are all files in the working directory that are not in the last snapshot and are not in the staging area.

Q14 - Git automatically adds new files to the repository and starts tracking them. True or False? Give reasons.
Sol: NO.
Git does not add them to the repository as you need to run (git add) command to add it and then push it to the repo.

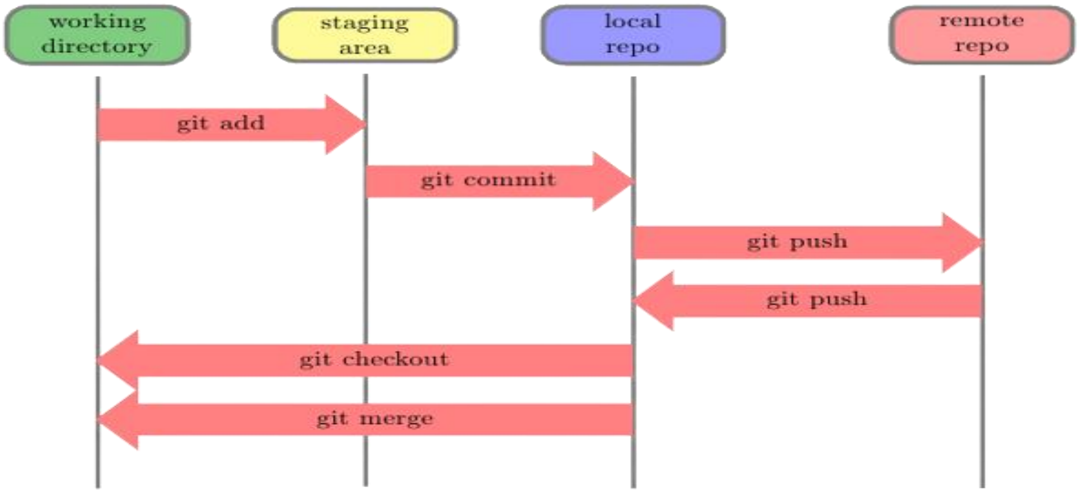
Q15 - What is the command to commit the staged changes for the Git repository?
Sol:

⇨ git commit

Q16 - What is the command to commit with the message "New email"?
Sol:

⇨ git commit -m " New email"

Q17 - Draw the full workflow of Git and describe the diagram.
Sol:



1. Working directory - where the user does all their work.
2. Staging Area – Where users list changes made to the working directory before creating (committing) a snapshot.
3. repository – where Git keeps these changes as different versions of the project.

Basic Workflow

1. Create a local git repository.To start tracking the project, type git init into your terminal.

Git Status This isn't necessarily a step in the process, but running git status will help to keep track of progress within the workflow.

2. Add file to the staging area When run git init, it doesn't mean that any changes are being recorded yet. To tell git to start tracking a file, you must run git add <filename>.

3. Commit your changes When your file is in the staging area.

To Review: The basic git workflow consists of creating a local repository (mkdir), adding that to the staging area (git add), and committing your changes (git commit).

Q18 - What is a branch in Git?
Sol: A branch is a different version of the repository than the main working project. This is a feature available in most modern version control systems.

- ⇨ A Git project can have multiple branches.
- ⇨ These branches are pointers to snapshots of changes.

Q19 - What is the command to create a new branch named "new-email"?
Sol: There are many ways to create a new branch:

- ⇨ git branch new-email

Q20 - What is the command to move to the branch named "new-email"?
Sol:

- ⇨ git checkout new-email

Q21 - What is the option, when moving to a branch, to create the branch it if it does not exist?
Sol: there ate 2 ways:

- ⇨ git checkout -b [branch name]
- ⇨ git switch -c [branch name]

Q22 - What does the git init command does?
Sol: Create a new, empty repository using the git init command.

- ⇨ Used to build an existing project as a git project.
- ⇨ Used to initialize.
- ⇨ The git init command creates a .git subdirectory in the current working directory. This newly created subdirectory contains all the required metadata

Q23 - What is a fork? How is it different from clone in Git? How do you fork and clone a repository?
Sol: **Fork** creates a completely independent copy of a Git repository.

- ⇨ Unlike forks, Git clones create linked copies that stay in sync with the target repository.

Steps to fork the repo:

1. On GitHub.com, navigate to the repository.

2. In the top-right corner of the page, click Fork.

3. Select an owner for the forked repository.

4. Optionally, add a description of your fork.

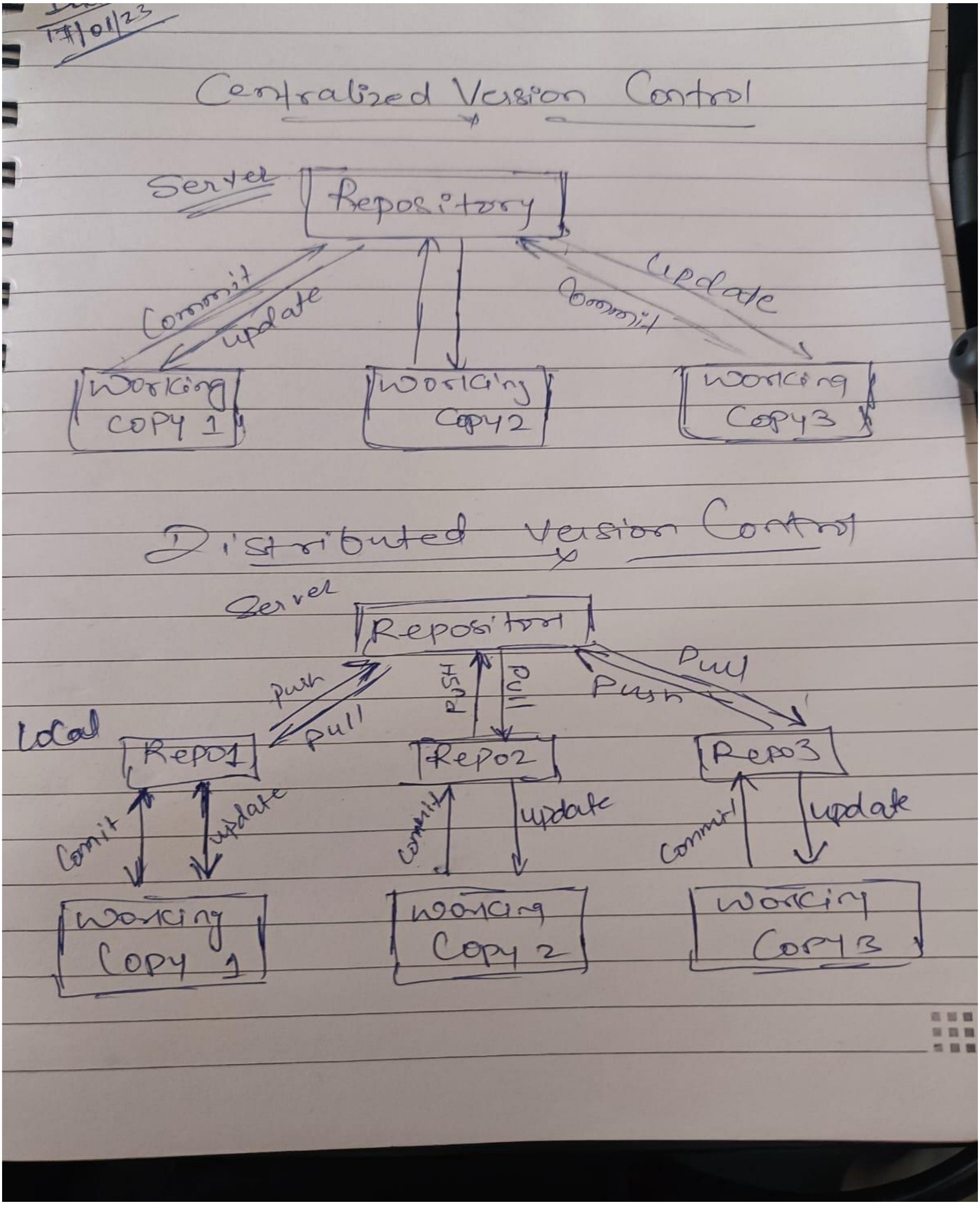
5. Create fork.

Q24 - What does 'push' mean in Git? Give the command.
Sol:
The term push refers to uploading the contents of your local repository to a remote repository.

- ⇨ Pushing is the act of transferring commits from your local repository to a remote repository.
- ⇨ Pushing may overwrite your changes.
- ⇨ git push origin [branch]

Q25 - Draw the standard architecture of two types of version control systems. (Hint - server project repo, working directories etc.) Explain the diagrams.

Sol:


17/01/23
Centralized Version Control
Server
Repository
Commit
update
update
Commit
Working Copy 1
Working Copy2
Working Copy3
Distributed Version Control
Server
Repository
push
pull
PUSH
Pull
Pull
Push
Local
Repo1
Repo2
Repo3
Commit
update
Commit
update
Commit
update
Working Copy 1
Working Copy 2
Working Copy3