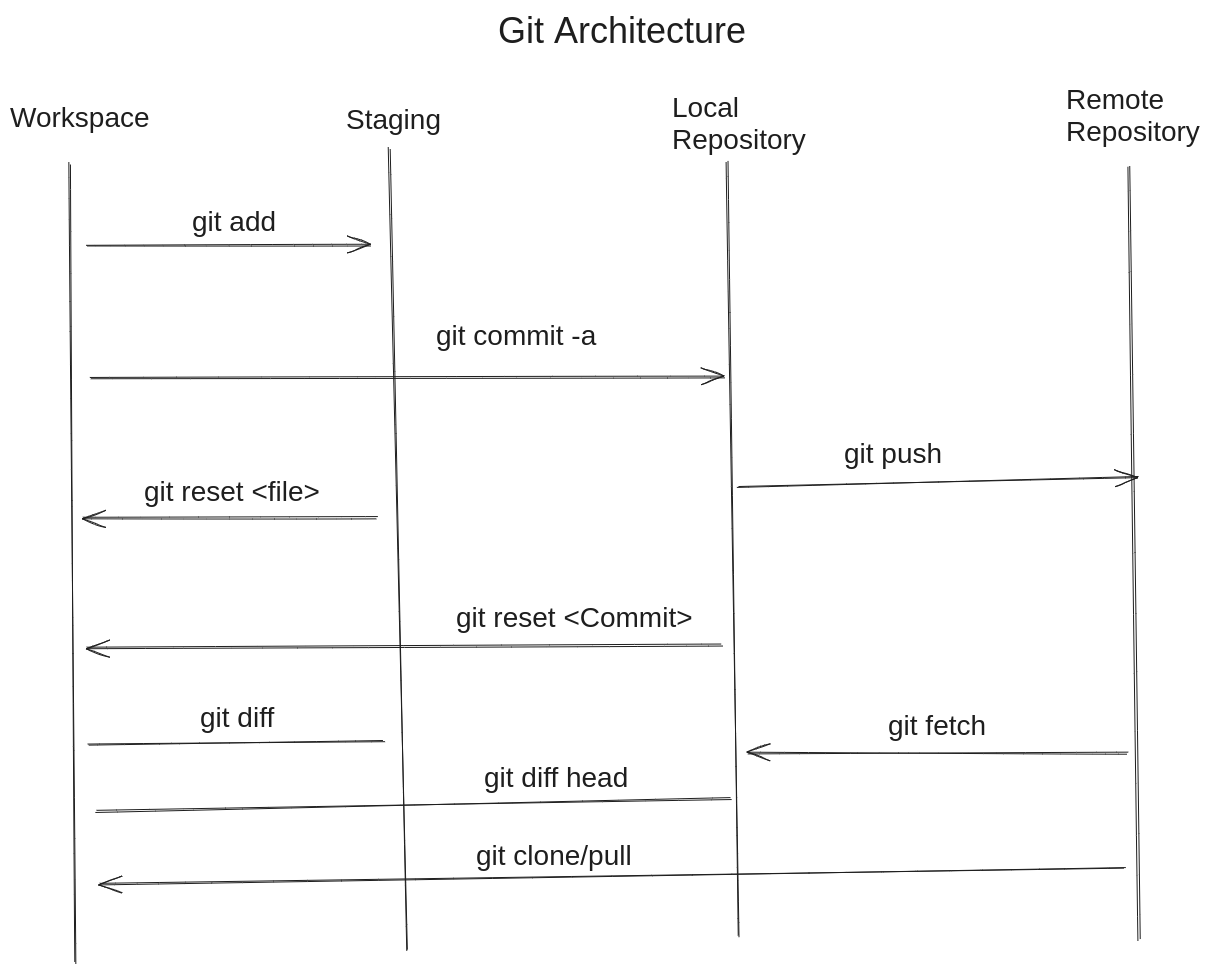
**1.Explain the Architecture of Git.**

The architecture of Git is designed to provide a flexible, scalable, and efficient way to manage source code changes. Git has a client-server architecture that allows multiple developers to work on the same codebase at the same time without the need of a centralised server. Each developer has a local copy of the entire repository, which allows them to work on the code offline and without need of the network connection.



Here are the key components of the git architecture:

1. **Git Repository:** A git repository is a collection of files and folders, along with the version history of those files. It’s managed using it, which is a distributed version control system.
2. **Commit:** A commit refers to a snapshot of changes to the files in a Git repository at a particular point in time. When there are changes in the project file, Git allows to record these changes by creating a commit.
3. **Branch:** A branch is a separate line of development. It allows you to work in different versions of your project simultaneously, without affecting the main or other branches until you decide to merge changes.
4. **Merge:** A merge is a process of combining changes from one branch into another branch. When two branches are diverged, a merge can be used to combine the changes from both branches into a single codebase.
5. **Remote:** A remote is a copy of the repository that is stored on a local machine on a different machine or server. Remote can be used to collaborate with other developers and to synchronise changes between different copies of the repository.
6. **Clone:** A clone is a copy of a repository that is stored on a local machine. Clones can be used to work on the database offline and without the need of the network connection. Clones can be created from a repository or from another local copy of the repository.
7. **Pull:** A pull is a process of downloading changes from a remote repository and merging them into the local copy of the repository.
8. **Push:** A push is a process of uploading changes from a local copy to the repository to a remote repository.

**2. Explain all the git commands.**

1. **git config:** Configure git settings, such as user information, preferences, and aliases.

**Example:** git config --global user.name "Your Name"’

Git config –global user.email “You email”

1. **git init:** Initialises new Git repository in the current directory.
2. **git clone:** Clone an existing repository from a server remote.

git clone <repository-url>

1. **git add:** Add file contents to the index (staging area) for the next commit.

git add <File> (add a specific file)

git add . or git add -A (add all changes)

1. **git status:** Show the status of files in the working directory and staging area.

git status

1. **git commit:** Record changes to the repository with a commit message.

git commit -m “Commit message”

1. **git branch:** List, create, or delete branches.

git branch (list branches)

git branch <branch-name> (create a new branch)

git branch -d <branch-name> (delete a branch)

1. **git checkout:** Switch branches and restore working tree files.

git checkout <branch-name> (switch to a branch)

git checkout -b <new-branch> (create and switch to a new branch)

1. **git merge:** Combine changes from one branch to another.

git merge <branch-name> (merge changes into current branch)

1. **git remote:** The ‘git remote’ command is used to manage remote connections to repositories. Remotes are pointers to repositories located elsewhere, such as on a server or another machine.

git remote add <name> <url> (add a remote repository)

git remote -v (list remotes)

1. **git push:** Update remote repository with local commits.

git push <remote-name> <branch-name>

1. **git pull:** Fetch from and integrate with another repository or a local branch.

git pull <remote-name> <branch-name> (fetch and merge)

1. **git log:** Show commit logs.

git log

1. **git diff:** Show changes between commits, commit and work tree, etc.

git diff (show changes in working directory)

git diff <commit1> <commit2> (show changes between commits)

1. **git reset:** The ‘git reset’ command is a powerful tool in Git that allows you to undo changes in your repository.

git reset --hard HEAD (discard all changes in working directory)

git reset --soft HEAD (undo the last commit, keeping changes)

1. **git revert:** ‘git revert’ is a Git command used to create a new commit that undoes the changes introduced by a previous commit. Unlike ‘git reset’ which can rewrite commit history, it is a safe way to undo changes because it adds a new commit to the history, preserving the original commit.

git revert <commit>

1. **git clean:** Remove untracked file from the working tree.

git clean -n (dry run to see what would be removed)

git clean -f (remove untracked files)

1. **git stash: ‘**git stash’ is a useful command that allows you to temporarily save (stash) changes that you don’t want to commit immediately. This is specially handy when you need to switch branches or perform other tasks without committing half-done work.

git stash (stash changes)

git stash apply (apply the latest stash)

**3. Write a step to create a new branch and merge with master branch**

**Step 1:** git branch <branch-name> (Branch created)

**Step 2:** git checkout <branch-name> (Switch to a branch)

**Step 3:** git checkout master (Switch to the master branch)

**Step 4:** git merge <branch-name> ( Merger the desired branch with the master branch)

**4. Explain fork and git clone with examples.**

**Fork:** In Git, a "fork" refers to creating a personal copy of someone else's repository. Forking a repository allows you to freely experiment with changes without affecting the original project. It is commonly used in open-source projects on platforms like GitHub, GitLab, and Bitbucket.

**git clone:** The git clone command is used to create a copy of a remote repository on your local machine. This command not only copies the files and directories of the repository but also copies the entire version history of the project.

Example: git clone <remote repository’s git clone url>