DAY 6 Assignment

Q)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 for a centralized server. Each developer has a local copy of the entire repository, which allows them to work on the code offline and without the need for a network connection.

Here are the key components of the Git architecture:

1. Git Repository: Git repository is the central data store for all of the files, directories, and project history in a project folder. In other words, this is the place where all of the changes that have been made to the codebase are stored. The repository can be located on a local machine, on a network server, or on a cloud-based platform.

2. Commit: A commit is a snapshot of the codebase at a particular point in time. It represents a complete set of changes made to the codebase, including new files, modifications to existing configuration files, edit files, and deleted file contents. Each commit has a unique identifier that can be used to reference it at any point in the future.

3. Branch: A branch is a separate line of development that is created from a specific commit in the codebase. Each Git branch can have its own set of changes and commits, which can be merged back into the main branch of the codebase when they are ready.

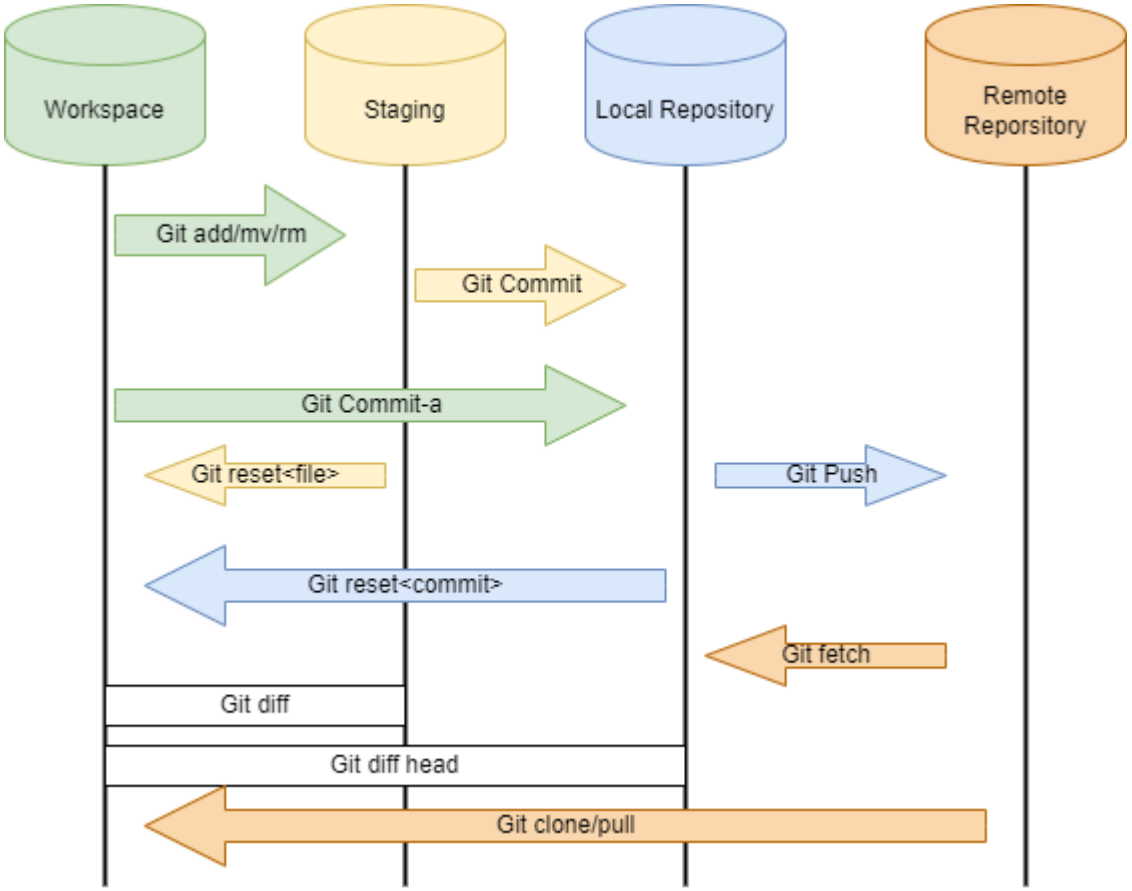
4. Merge: A merge is a process of combining changes from one branch into another branch. When two branches have diverged, a merge can be used to combine the changes from both branches into a single codebase.

5. Remote: A remote is a copy of the repository that is stored on a different machine or server. Remotes can be used to collaborate with other developers and to synchronize changes between different copies of the repository.

6. Clone: A clone is a copy of the repository that is stored on a local machine. Clones can be used to work on the codebase offline and without the need for a network connection. Clones can be created from a remote repository or from another local copy of the repository.

7. Pull: A pull is the 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 of the repository to a remote repository.



Q)2) Explain all git commands:

a) Git config: This command configures the user.This command sets the author name and email address to be used with your commits.

Syntax:

$ git config --global user.name "ImDwivedi1"

$ git config --global user.email "Himanshudubey481@gmail.com"

b) Git Init: This command is used to create a local repository.

Syntax:

$ git init Demo

c) Git clone: This command is used to make a copy of a repository from an existing URL. If I want a local copy of my repository from GitHub, this command allows creating a local copy of that repository on your local directory from the repository URL.

Syntax:

$ git clone <URL>

d) Git add: This command is used to add one or more files to staging (Index) area.

Syntax:

To add one file

$ git add <Filename>

To add more than one file

$ git add*

e)Git commit: Commit command is used in two scenarios. They are as follows.

Git commit -m: This command changes the head. It records or snapshots the file permanently in the version history with a message.

Syntax:

$ git commit -m " Commit Message"

Git commit -a: This command commits any files added in the repository with git add and also commits any files you've changed since then.

Syntax:

$ git commit -a

f)Git status: The status command is used to display the state of the working directory and the staging area. It allows you to see which changes have been staged, which haven't, and which files aren?t being tracked by Git.It also lists the files that you've changed and those you still need to add or commit.

Syntax:

$ git status

g) Git push: It is used to upload local repository content to a remote repository. Pushing is an act of transfer commits from your local repository to a remote repo.

Git push -all: This command pushes all the branches to the server repository.

Syntax:

$ git push --all

h) Git pull: Pull command is used to receive data from GitHub. It fetches and merges changes on the remote server to your working directory.

Syntax:

$ git pull <URL>

i) Git Branch: This command lists all the branches available in the repository.

Syntax:

$ git branch

j) Git Merge: This command is used to merge the specified branch?s history into the current branch.

Syntax:

$ git merge BranchName

k) Git log: This command is used to check the commit history.

Syntax:

$ git log

l) Git remote: Git Remote command is used to connect your local repository to the remote server. This command allows you to create, view, and delete connections to other repositories.

M) Git tag: This command use to list all tags.

Syntax:

$ git tag

N) Git diff: Shows changes between commits, commit and working tree, etc.

Syntax:

$ git diff

Steps to Create a New Branch and Merge with the Master Branch

Step 1: Create a New Branch

a) Switch to the master branch and ensure it is up-to-date:

$ git checkout master

$ git pull origin master

b) Create and switch to the new branch:

$ git checkout -b <new_branch_name>

Step 2: Work on the New Branch

c) Make your changes in the new branch.

d) Stage the changes:

```
$ git add <file_name>
```

e) Commit the changes:

```
$ git commit -m "Description of changes"
```

Step 3: Merge the New Branch with the Master Branch

f) Switch back to the master branch:

```
$ git checkout master
```

g) Ensure the master branch is up-to-date:

```
$ git pull origin master
```

h) Merge the new branch into the master branch:

```
$ git merge <new_branch_name>
```

Step 4: Push Changes to the Remote Repository

i) Push the updated master branch to the remote repository:

```
$ git push origin master
```

What is a Fork?

A fork is a copy of a repository that you manage. Forks allow you to freely experiment with changes without affecting the original project. Most commonly, forks are used to propose changes to someone else's project or to use someone else's project as a starting point for your own idea.

Example of Forking a Repository

1)Go to the Repository on GitHub:

Visit the GitHub repository page that you want to fork. For example, if you want to fork a repository called example-repo located at https://github.com/original-owner/example-repo.

2)Fork the Repository:

Click the "Fork" button at the top-right corner of the repository page. This creates a copy of the repository under your GitHub account. For example, if your GitHub username is your-username, the forked repository will be located at https://github.com/your-username/example-repo.

3)Clone Your Fork:

Now, you can clone the forked repository to your local machine for development.

$ git clone https://github.com/your-username/example-repo

4)Add the Original Repository as a Remote:

To keep your fork in sync with the original repository, add the original repository as a remote.

$ cd example-repo

$ git remote add upstream https://github.com/original-owner/example-repo

5)Fetch Updates from the Original Repository:

Periodically, fetch and merge updates from the original repository.

$ git fetch upstream

$ git merge upstream/master

What is Git Clone?

Git clone is a command used to create a local copy of a remote repository. Cloning a repository creates a complete copy of the repository, including all branches, commits, and files.

Example of Cloning a Repository

1)Find the Repository URL:

Obtain the URL of the repository you want to clone. For example, https://github.com/original-owner/example-repo.

2)Clone the Repository:

Use the git clone command followed by the repository URL to clone it to your local machine.

$ git clone https://github.com/original-owner/example-repo

3)Navigate to the Cloned Repository:

After cloning, navigate into the repository directory.

$ cd example-repo

4) Start Working

Now have a complete local copy of the repository and can start making changes.