**GIT (Global Information Tracker)**

Git config: This command sets the author’s name and email address respectively to be used with your commits. Git allows you to customize and set various configuration options like.

Git config --global user.name “saibaji”

Git config –global user. Email [bajisai77@gmail.com](mailto:bajisai77@gmail.com)

Git in it:

Init is short for initialization and is a term used in computing to refer to process of setting up a system. This command is used to start a new repository.

Git init

Git clone:

Git clone Is primary used to point to an existing repo and make a clone or copy of the repo and a new dictionary at the anther location.

Git clone https://github.com/udaykishore234/git2.git

Git add add.py:

Git add commend is used to move changes made to file in your working directory to staging area.

Git add.

Git commit:

Git commit is a snapshot of your project at a specific point in the time.

Git commit -m “this is my commit”

Git diff:

Diffing is a function that takes two input date sets and output the changes between them.

Git diff

Git reset:

Reset is the commend we use when we want to move the repository back to previous commit.

Git reset

Git rm:

Git used to remove a file a git repository.

Git remove -u origin file name

Git log:

Git log commend the commit history to git repository.

Git log

Git show:

Git show displays details about a specific commit in a git repository history.

Git show

Git tag :

Used to create a label or marker on a specific commit in your repository

Git tag

Git branch:

Create a tag on a specific branch

Git branch -m main

Git checkout:

Used to switch between different branch with in a repository.

Let’s you navigate between the branches created by get branch.

Git merge:

Gits way of putting a forked history back together again.

Git merge name

Git remote:

Allows you to mange these connections, adding, new remotes, viewing existing ones or deleting them.

Git remote -m main

Git push:

Used to upload local changes from your repository to a remote repository.

Git push -u origin main

Git pull:

Used to fetch and download content from a remote repository and immediately upload the local repository to match that content.

Git pull name

Git stash:

A feature that allows to temporary save your committed changes to your working directory.

Git stash apply

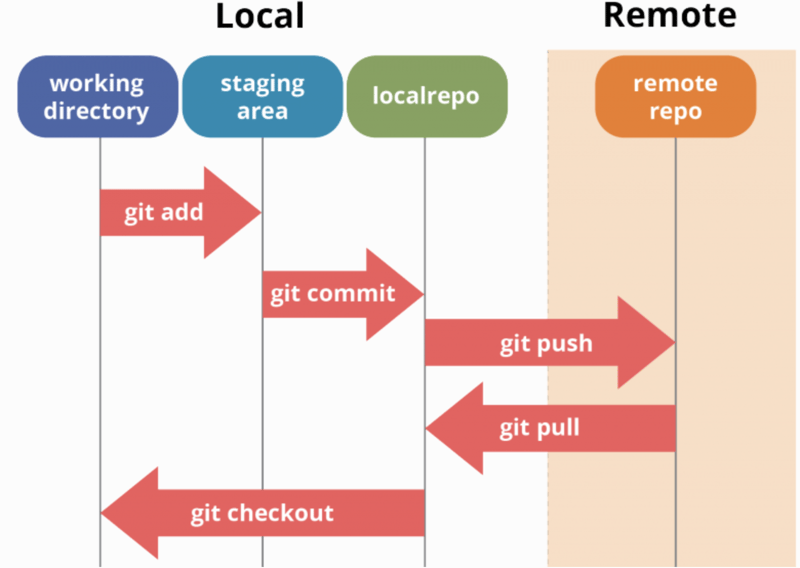
**1.Explain the workflow of git with neat diagram**

The typical git workflow involves three key stages

Working, directory, staging area, and local repository.

Git is a distributed version control system that helps developers track changes in their codebase.

1. **Working Directory** – This is where you modify files in your project.
2. **Staging Area (Index)** – Before committing changes, you add them to the staging area to prepare them for the next commit.
3. **Local Repository (. git folder)** – Once committed, changes are stored in the local repository.
4. **Remote Repository (GitHub, GitLab, etc.)** – Your local commits can be pushed to a remote repository for collaboration.



Git init → Initializes a new Git repository.

Git status → Shows the status of changes in the working directory.

Git add <filename> → Adds specific files to staging.

Git add . → Adds all modified files to staging.

Git commit -m "Commit message" → Saves changes in the local repository.

Git log → Displays the commit history.

Git it remote add origin <repository URL> → Links the local repo to a remote one.

Git push origin main → Uploads local commits to the remote repository.

Git pull origin main → Fetches and merges remote changes into the local branch.

Git branch <branch name> → Creates a new branch.

Git checkout <branch name> → Switches to a different branch.

Git merge <branch name> → Merges changes from another

branch into the current branch.

**2. What is Git? How to Stage a File with git add?**

* **Git**: A distributed version control system that tracks changes in source code during software development.
* **To stage a file**, use

git add filename

git add .

**3. What is Git, GitHub, and GitLab? Explain the Differences**

| **Feature** | **Git** | **GitHub** | **GitLab** |
| --- | --- | --- | --- |
| Definition | Version control system | Cloud-based Git repository | Git-based DevOps platform |
| Hosting | Local or Remote | Hosted | Self-hosted & Hosted |
| CI/CD | No built-in | Requires GitHub Actions | Built-in |
| Access | CLI-based | Web-based UI | Web-based UI |

**4. How to Create a Tag and Push It**

* Create an **annotated** tag:

git tag -a v1.0 -m "Version 1.0 release"

* Create a **lightweight** tag:

git tag v1.0

* Push a tag to the remote repository:

git push origin v1.0

* Push all tags:

git push origin --tags

**5. Steps to Send a Project from Local Repo to Remote Repo**

1. Initialize Git:

git init

1. Add remote URL:

git remote add origin <repo\_url>

1. Add files to staging:

git add.

1. Commit the changes:

git commit -m "Initial commit"

1. Push to the remote repository:

git push -u origin main

**6. What is a Pull Request? Explain Its Procedure**

A **pull request (PR)** allows developers to propose changes from one branch to another before merging.

**Steps to Create a PR:**

1. **Fork a repository** (if contributing to someone else's project).
2. **Clone the repository** and create a feature branch:

git checkout -b feature-branch

1. **Make changes, commit, and push**:

git add .

git commit -m "Added feature"

git push origin feature-branch

1. **Create a pull request** on GitHub/GitLab and request a review.
2. **Review and merge** after approval.

**7. What is a Fork? Why Is It Necessary?**

* **Forking** creates a personal copy of a remote repository under your account.
* It is **necessary** when you want to contribute to someone else's project without affecting the original repository.

**Steps to Fork and Contribute:**

1. Click **Fork** on GitHub/GitLab.
2. Clone your forked repo:

git clone <your-fork-url>

1. Create a feature branch, make changes, and push.
2. Open a **Pull Request** to the original repo.

Top of Form

Bottom of Form