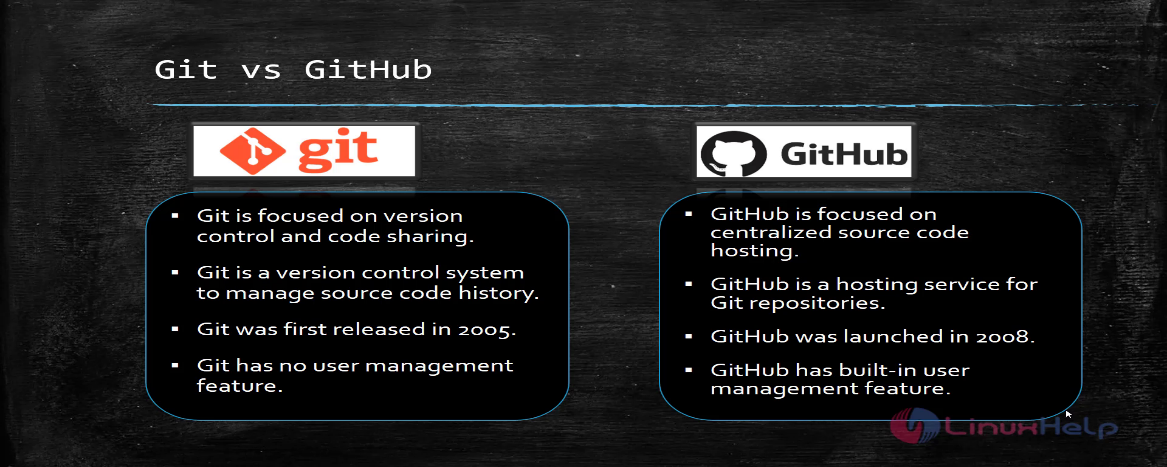
**GIT :**

Git is a distributed version control system that is widely used for tracking changes in source code during software development.

**GitHub:**

* GitHub is a web-based platform built around Git.
* It provides hosting for Git repositories and offers additional features for collaboration, such as issue tracking, pull requests, code review tools, and project management.
* GitHub allows developers to share their code, contribute to open-source projects, and collaborate with others on software development projects.

**Difference between Git and GitHub :**



**Git Commands :**

1. **git init**: Initialize a new Git repository in the current directory.
2. **git clone <repository URL>**: Clone a remote repository to your local machine.
3. **git add <file>**: Add a file or directory to the staging area for the next commit.
4. **git commit -m "<commit message>"**: Commit staged changes with a descriptive message.
5. **git status**: Check the status of the working directory and staging area.
6. **git pull**: Fetch and merge changes from the remote repository into the current branch.
7. **git push**: Push commits from your local repository to the remote repository.
8. **git branch**: List, create, or delete branches.
9. **git checkout <branch>**: Switch to a different branch.
10. **git merge <branch>**: Merge changes from a specified branch into the current branch.

To add a file to a GitHub repository using Git commands, follow these steps:

1. **Clone the Repository(if not already cloned)**:

**Bash 🡪git clone <repository URL>**

Replace <repository URL> with the URL of the GitHub repository you want to clone.

1. **Navigate to the Cloned Repository**:

**cd<Repository Name>**

Replace **<repository name>** with the name of the directory that was created when you cloned

the repository.

1. **Check the Status**:

**git status**

Before adding files, it's a good practice to check the status of your repository to see which files have been modified and are ready to be staged.

1. **Add Files to the Staging Area**:

**git add <file name>**

Use the **git add** command followed by the names of the files you want to stage. You can specify individual files or directories. For example, to stage a single file.

**Or to stage all modified files in the current directory and its subdirectories:**

**git add .**

1. **Commit Changes**:

**git commit-m “your commit message here”**

After staging the files, you need to commit them to create a new snapshot in the repository. Use the **git commit** command with a descriptive message.

1. **Push the Changes to GitHub**:

**git push origin <branch name>**

you may want to push your committed changes to the remote repository. Use the **git push** command.

Replace **<branch name>** with the name of the branch you want to push the changes to, typically **main** or **master**.