What is Git?

Git is a tool that helps developers keep track of changes in their code and work together on projects. It allows you to save versions of your project, so you can go back to an earlier version if needed.

Key Points:

- **Distributed** – Everyone working on the project has a copy of the entire code.
- **Branching** – You can work on different parts of the project without affecting the main project.
- **Version Control** – Keeps track of changes and lets you undo them if something goes wrong.
- **Collaboration** – Multiple people can work on the same project at the same time.

What does Git do?

- **Track Changes**: Git saves every change made to the code, so you can see what was changed and who changed it.
- **Version Control**: Git creates snapshots (versions) of your project, so you can go back to an earlier version if something goes wrong.
- **Work with Others**: Git makes it easy for multiple people to work on the same project at the same time without interfering with each other's work.
- **Branching**: You can create branches to work on new features or fixes without affecting the main project. Once the work is done, you can merge it back into the main project.

Why Git?

- Over 70% of developers use Git!
- Developers can work together from anywhere in the world.
- Developers can see the full history of the project.
- Developers can revert to earlier versions of a project.

What is GitHub?

GitHub is a web-based platform that uses **Git** for version control. It allows developers to store their code online, collaborate with others, and track changes to their projects.

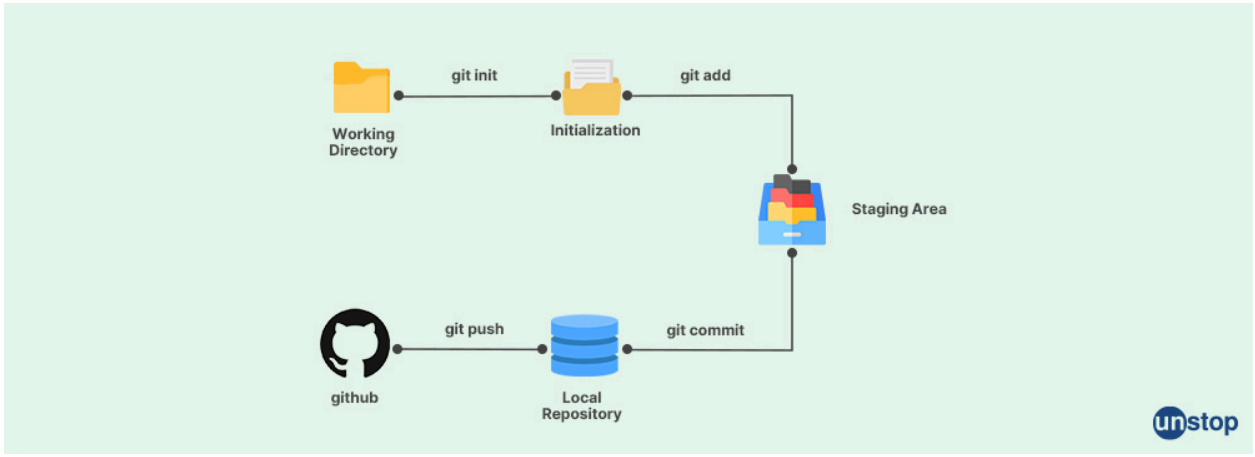
Key Features of GitHub:

1. **Repository Hosting**: It provides cloud storage for your Git repositories, so you can easily share and access your code from anywhere.
2. **Collaboration**: Multiple developers can work on the same project, suggest changes, and merge their work together.
3. **Version Control**: It integrates with Git to track changes and manage versions of your project.

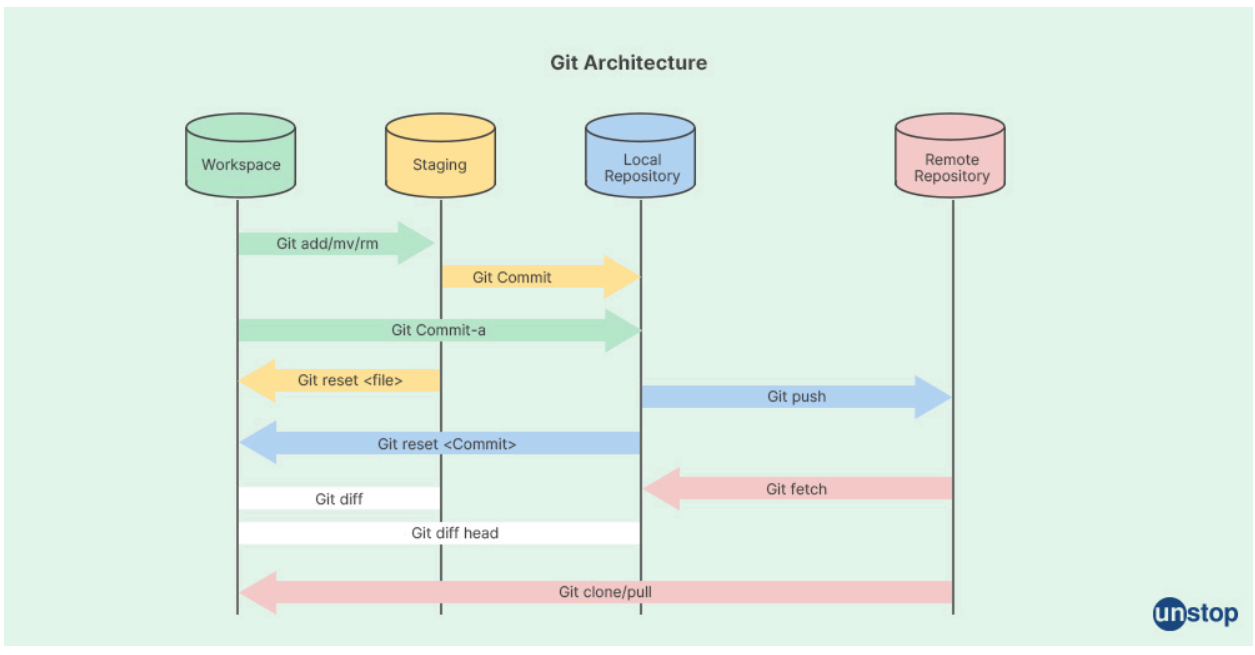
Difference between Git and GitHub :-

Git	GitHub
A version control system for tracking code changes.	A web-based platform that hosts Git repositories and supports collaboration.
Used for managing code locally and tracking changes.	Used for storing, sharing, and collaborating on code online.
A tool/command-line software.	A cloud-based service for Git repositories.
Installed on your local machine.	No installation needed; web-based platform.
Does not provide cloud hosting.	Hosts code repositories online.
Limited to local collaboration.	Supports multiple developers working on the same project.
Manages versions locally.	Works with Git to manage and sync versions online.
Git itself is open-source.	GitHub is a platform that hosts open-source projects but is not fully open-source.
No cloud backup; everything is local.	Provides cloud backup of your repositories.
Used for tracking and managing code versions locally.	Used for sharing code, collaboration, and version control in the cloud.

The Git Lifecycle



Git Architecture :-



Steps for Git :-

```
MINGW64:/c/Users/Admin/Desktop/GITDEMO/WAD-Assingments

Admin@DESKTOP-7CAIK3B MINGW64 ~
$ pwd
/c/Users/Admin

Admin@DESKTOP-7CAIK3B MINGW64 ~
$ git config --global user.name "SamruddhiPote"

Admin@DESKTOP-7CAIK3B MINGW64 ~
$ git config --global user.email "samruddhirpote@gmail.com"

Admin@DESKTOP-7CAIK3B MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=main
user.name=SamruddhiPote
user.email=samruddhirpote@gmail.com

Admin@DESKTOP-7CAIK3B MINGW64 ~
$ cd Desktop

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop
$ cd GIT DEMO
bash: cd: too many arguments

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop
```

```
MINGW64:/c/Users/Admin/Desktop/GITDEMO/WAD-Assingments
Admin@DESKTOP-7CAIK3B MINGW64 ~
$ cd Desktop

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop
$ cd GIT DEMO
bash: cd: too many arguments

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop
$ cd GITDEMO

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO
$ git clone https://github.com/SamruddhiPote/WAD-Assingments.git
Cloning into 'WAD-Assingments'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO
$ cd WAD-Assingments/

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO/WAD-Assingments (main)
$ cd WAD-Assingments
bash: cd: WAD-Assingments: No such file or directory

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO/WAD-Assingments (main)
$ cd ..

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO
$ cd WAD-Assingments

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO/WAD-Assingments (main)
$ ls
README.md

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO/WAD-Assingments (main)
$ ls -a
./  ../  .git/  README.md
```

```
MINGW64:/c/Users/Admin/Desktop/GITDEMO/WAD-Assingments

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO/WAD-Assingments (main)
$ git add README.md

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO/WAD-Assingments (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
        new file:   index.html


Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO/WAD-Assingments (main)
$ git commit -m "Added New File(index.html)"
[main f6be5f3] Added New File(index.html)
 2 files changed, 3 insertions(+), 1 deletion(-)
 create mode 100644 index.html

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO/WAD-Assingments (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO/WAD-Assingments (main)
$ git push origin main
info: please complete authentication in your browser...
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 369 bytes | 369.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
```


```
MINGW64:/c/Users/Admin/Desktop/GITDEMO/WAD-Assingments
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
        new file:   index.html


Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO/WAD-Assingments (main)
$ git commit -m "Added New File(index.html)"
[main f6be5f3] Added New File(index.html)
 2 files changed, 3 insertions(+), 1 deletion(-)
 create mode 100644 index.html

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO/WAD-Assingments (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO/WAD-Assingments (main)
$ git push origin main
info: please complete authentication in your browser...
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 369 bytes | 369.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
```


```
To https://github.com/SamruddhiPote/WAD-Assingments.git
   3aae501..f6be5f3  main -> main

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO/WAD-Assingments (main)
$ git pull
Already up to date.

Admin@DESKTOP-7CAIK3B MINGW64 ~/Desktop/GITDEMO/WAD-Assingments (main)
$
```