

Day – 4

Date:29/02/2024

To Do.

- What is GitHub & Git
- Git Commands

What is GitHub & Git ?

Git is a version control system that allows developers to track changes in their code. GitHub is a web-based hosting service for git repositories. In simple terms, you can use git without GitHub, but you cannot use GitHub without Git.

Git	GitHub
1. It is a software	1. It is a service
2. It is installed locally on the system	2. It is hosted on Web
3. It is a command line tool	3. It provides a graphical interface
4. It is a tool to manage different versions of edits, made to files in a git repository	4. It is a space to upload a copy of the Git repository
5. It provides functionalities like Version Control System Source Code Management	5. It provides functionalities of Git like VCS, Source Code Management as well as adding few of its own features

What is Git:

Git is a popular version control system. It was created by Linus Torvalds in 2005, and has been maintained by Junio Hamano since then.

It is used for:

- Tracking code changes
- Tracking who made changes
- Coding collaboration

What does Git do?

- Manage projects with **Repositories**
- **Clone** a project to work on a local copy
- Control and track changes with **Staging** and **Committing**
- **Branch** and **Merge** to allow for work on different parts and versions of a project
- **Pull** the latest version of the project to a local copy
- **Push** local updates to the main project

Working with Git

- Initialize Git on a folder, making it a **Repository**
- Git now creates a hidden folder to keep track of changes in that folder
- When a file is changed, added or deleted, it is considered **modified**
- You select the modified files you want to **Stage**
- The **Staged** files are **Committed**, which prompts Git to store a **permanent** snapshot of the files
- Git allows you to see the full history of every commit.
- You can revert back to any previous commit.
- Git does not store a separate copy of every file in every commit, but keeps track of changes made in each commit!

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?

- Git is not the same as GitHub.
- GitHub makes tools that use Git.
- GitHub is the largest host of source code in the world, and has been owned by Microsoft since 2018.
- In this tutorial, we will focus on using Git with GitHub.

Git commands :

\$ git init -

Initializes a new Git repository in the current directory. This command creates a hidden subfolder within your existing project that houses the internal data structure required for version control.

\$ git add . or git add <filename> -

\$ git status -

\$ git commit -m "message" -

\$ git remote add origin <link repo>

\$ git push -u origin main

* These are the generally used Git Commands to push the file in Repository

Git Commands :

Command	Description
git init	Initialize a local Git repository
git clone repo_url	Clone public repository
git clone ssh://git@github.com/[username]/[repository-name].git	Clone private repository
git status	Check status
git add [file-name]	Add a file to the staging area
git add -A	Add all new and changed files to the staging area
git commit -m "[commit message]"	Commit changes
git rm -r [file-name.txt]	Remove a file (or folder)
git branch	List of branches (the asterisk denotes the current branch)
git branch -a	List all branches (local and remote)
git branch [branch name]	Create a new branch
git branch -d [branch name]	Delete a branch
git branch -D [branch name]	Delete a branch forcefully
git push origin --delete [branch name]	Delete a remote branch
git checkout -b [branch name]	Create a new branch and switch to it
git checkout -b [branch name] origin/[branch name]	Clone a remote branch and switch to it
git branch -m [old branch name] [new branch name]	Rename a local branch
git checkout [branch name]	Switch to a branch
git checkout -	Switch to the branch last checked out
git checkout -- [file-name.txt]	Discard changes to a file
git merge [branch name]	Merge a branch into the active branch
git merge [source branch] [target branch]	Merge a branch into a target branch
git stash	Stash changes in a dirty working directory
git stash clear	Remove all stashed entries
git push origin [branch name]	Push a branch to your remote repository
git push -u origin [branch name]	Push changes to remote repository (and remember the branch)
git push	Push changes to remote repository (remembered branch)
git push origin --delete [branch name]	Delete a remote branch
git pull	Update local repository to the newest commit
git pull origin [branch name]	Pull changes from remote repository
git remote add origin ssh://git@github.com/[username]/[repository-name].git	Add a remote repository

git remote set-url origin ssh://git@github.com/[username]/[repository-name].git	Set a repository's origin branch to SSH
git log	View changes

Scribe:

[9:35 AM] Dibya (Guest)

Please open the link :

[9:35 AM] Dibya (Guest)

<https://softwaredominos.com/home/software-design-development-articles/high-level-solution-design-documents-what-is-it-and-when-do-you-need-one/#4-highlevel-design-hld>

[9:38 AM] Dibya (Guest)

PRD == BRD

[9:42 AM] Dibya (Guest)

<https://www.docker.com/>

[9:43 AM] Dibya (Guest)

<https://kubernetes.io/>

[9:44 AM] Dibya (Guest)

<https://aws.amazon.com/elasticbeanstalk>

[9:45 AM] Dibya (Guest)

<https://aws.amazon.com/pm/lambda/>