Open Code Assignment

Difference between Git and Github

Git is a free, open-source software distributed version control system (DVCS) designed to manage all source code history. It can keep a history of commits, can reverse changes, and lets developers share code. Each developer must have Git installed on his or her local device to collaborate. It is commonly referred to as one of the best DevOps tools to understand and use in the developer space, and it's among the most widely used tools today. Companies like Amazon, Facebook, and Microsoft use it, to name a few.

GitHub, on the other hand, is a web-based hosting service for Git repositories. It offers all of Git's DVCS SCM and has some additional features. This includes collaboration functionality like project management, support ticket management, and bug tracking. With GitHub, developers can share their repositories, access other developers' repositories, and store remote copies of repositories to serve as backups.

S.no	Git	GitHub
1.	Git is software.	It is a service.
2.	Linux maintains Git.	Microsoft maintains GitHub.
3.	It is a command-line tool.	It is a graphical user interface.
4.	You can install it locally on the system.	It is hosted on the web. It is exclusively cloud-based.
5.	It is a VCS to manage source code history.	It is a hosting service for Git repositories.
6.	It focuses on code sharing and version control.	It focuses on centralized source code hosting.
7.	It lacks a user management feature.	It has a built-in user management feature.
8.	Git was launched in 2005.	GitHub was released in 2008.
9.	Git has minimum external tool configuration.	It has an active marketplace for tool integration.
10.	It is open-source licensed.	It has a free-tier and pay-for-use tier.

Git Commands

1.**git config --global user.email "[valid-email]"**

-set an email address that will be associated with each history marker.

2. **git clone [url]**

- retrieve an entire repository from a hosted location via URL.

3. **git status**

-show modified files in working directory, staged for your next commit.

4. **git add [file]**

-add a file as it looks now to your next commit (stage).

5. **git reset [file]**

- unstage a file while retaining the changes in working directory.

6**. git commit -m "[descriptive message]"**

-commit your staged content as a new commit snapshot.

7.**git stash**

-Save modified and staged changes

8. git stash list

-list stack-order of stashed file changes

9.git stash pop

-write working from top of stash stack

10.git stash drop

-discard the changes from top of stash stack

Creating the pull request

1. On GitHub.com, navigate to the main page of the repository.
2. In the "Branch" menu, choose the branch that contains your commits.
3. Above the list of files, click Pull request.
4. Use the *base* branch dropdown menu to select the branch you'd like to merge your changes into, then use the *compare* branch drop-down menu to choose the topic branch you made your changes in.
5. Type a title and description for your pull request.
6. To create a pull request that is ready for review, click Create Pull Request. To create a draft pull request, use the drop-down and select Create Draft Pull Request, then click Draft Pull Request. For more information about draft pull requests, see "About pull requests."
7. After your pull request has been reviewed, it can be merged into the repository.