What is GitHub?When was it created? Why? By who? What similar platforms exist? Why would you use such a platform?

Answer :

Git is a distributed version-control system for tracking changes in source code during software development, whereas Github is code sharing and version control **service** for programmers to **remotely** **collaborate** with multiple people on the project. It offers all the distributed version control and source code management (SCM) functionality of Git. It also provides features like bug tracking, feature requests, task management.

Github was created to facilitate reliable hosting of Gits online and easy collaboration. Github aims to offer free and inexpensive hosting of code.

Github was founded in 2008 by Tom Preston-Werner, Chris Wanstrath, P. J. Hyett and Scott Chacon.

There exist many code hosting platforms like CVS (1986), Subversion (2000), Mercurial, BitKeeper.

One would use Github for better and easy software configuration management.

As the source code is a valuable asset, one would use Github to protect code from physical damages or human errors.

It can also be used synchronize work of multiple developers without wasting hours merging.

It can also be used to find when a bug was introduced easily.

It can be used to see the entire timeline of the code changes, decisions, and progression of any project in one place.

* Repository

A repository encompasses the entire collection of files and folders associated with a project, along with each file’s revision history.

The file history appears as snapshots in time called commits, and the commits exist as a linked-list relationship, and can be organized into multiple lines of development called branches.

* Commit

Git commit saves the snapshot to project history and completes the change tracking process. Anything that is staged by ‘git add’ will become part of snapshot once ‘git commit’ is used.

* Push

This is used to push the local changes/commits to remote repository so that changes are available for all other collaborating members

* Branch

Branching of a project helps to try out new ideas/features without affecting the master branch. This gives us freedom to experiment and commit changes, safe in the knowledge that our branch won't be merged until it's ready to be reviewed by someone we are collaborating with.

* Fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Most commonly, forks are used to either propose changes to someone else's project or to use someone else's project as a starting point for your own idea.

A connection always exists between the fork and the original repository itself so that one can contribute back to original project using pull requests.

* Merge

Merge is used when we work on branch and is now ready to be merged into master branch. To merge changes to master, first we would have to check out the master and then run merge command to merge the changes to master

* Clone

Cloning also lets you make local copy of the repository but its ideal for instances when you need a way to quickly get your own copy of a repository where you may not be contributing to the original project.

* Pull

This is used to pull changes from remote repository into personal local repository

* Pull request

Pull requests are made to another repository to ask their maintainers to pull your changes into theirs that is Pull requests let you tell others about changes you've pushed to a branch in a repository on GitHub. Once a pull request is opened, you can discuss and review the potential changes with collaborators and add follow-up commits before your changes are merged into the base branch.