**THE NEED FOR GITHUB**

* Developers need a web/ cloud-based code hosting platform.
* Useful for version control.
* Enables effective collaboration.
* Download projects and files in one go.
* Easy evaluation of each other’s work.

Code hosting services that lets you manage repositories.

**GITHUB**

* Immensely powerful community.
* The largest shared repository.
* Easy Version Control.
* Secure cloud storage.

**What is GitHub?**

* Web-based git repository hosting service.
* Easy management of code.
* Open-source software for version control.
* Effective collaboration.
* Bug tracker

|  |  |
| --- | --- |
| **GIT**   * Git is a version control system, a tool to manage your source code history. * Installed and maintained in your local system. * Git is the tool | **GITHUB**   * GitHub is a hosting service for git repositories. * Exclusively cloud-based. * GitHub is the service for projects that use git. |

**WHAT IS GITHUB?**

* Git pushes or pulls data from the central server.
* GitHub is a core hosting platform for version control collaboration.
* It allows individuals and teams to host and share their code repositories and track changes to their code over time.
* GitHub is a company that allows you to host a central repository in a remote server.

**Difference between GitHub repository and Project:**

|  |  |
| --- | --- |
| **REPOSITORY**   * It can belong to users or organizations. * It is the actual storage space for code and files. * It can have multiple projects. * 1 repository = 1 or many projects. | **PROJECT**   * It can belong to repository or organizations. * It is a tool for managing and organizing the work that happens within the repository. * It can only be associated with one repository at a time. * 1 project = many repositories |

**Create a repository**

* Repository is the storage space for your project.
* GitHub is a very popular central repository that allows you to share your files.
* Push your local repository into github and share it with other collaborators via the central repository.

**Repository**

* It is the most basic element of GitHub.
* They are easiest to imagine as a project’s folder.
* A repository contains all of the project files (including documentation) and stores each files revision history.
* Repository can have multiple collaborators and can be either public or private.

**Project**

* GitHub Projects is a tool for managing tasks and organizing work within a repository.
* Project boards on GitHub help you organize and prioritize your work.
* It allows developers to create boards and cards that represent tasks, issues, or features that need to be completed.
* You can create project boards for specific feature work, comprehensive roadmaps, or even release checklists, with project boards, you have flexibility to create customized workflows that suit your needs.

**Steps to create a repository on GitHub:**

1. Log in to your GitHub account.
2. Click on the “+” icon in the top-right corner of the screen and select “New repository” from the drop-down menu.
3. Choose a name for your repository and add a short description of the project.
4. Decide whether you want the repository to be public or private.
5. Choose whether you want to initialize the repository with a README file or not.
6. Choose a license if you want to include one.
7. Click on the “Create repository” button.

Once you have created your repository, you can start adding files, making changes, and collaborating with others. You can also customize the repository settings, such as adding collaborators, setting up branches, and configuring webhooks.

**CREATE A BRANCH**

* Branches allow you to work on other features.
* They can be included with the main line of your project.
* The main branch is the one where all changes eventually get merged back into and is called master.

**MAKE A COMMIT**

* Records changes to one or more files in your branch.
* Git assigns each commit a unique ID, called a SHA or hash, that identifies the specific changes.

**OPEN AND MERGE PULL REQUESTS**

* Notifies developers about changes you’ve pushed to a branch in a repository.
* Acknowledge and review changes.
* You can create a pull request on github in two ways.
* Using GitHub web interface
* Using Command Line Interface CLI

Once the pull request has been created, it will be reviewed by the repository owner or maintainers, and they can accept or reject the changes as needed.

**SVC** – Software Version Control.

**SCM** – Source Code Management.

Both SVC and SCM are same. Examples: Git, Subversion, Mercurial, Perforce