Version Control with Git

**Git?**

Git is one the most used web-based repository. A web-based repository means a central place where all the project codes resides. Developers used to store their everyday changes, share, tests and collaborate among others for codes. Git is distributed version control system which will manager any changes in local files among many people. Git is a free and open source distributed version control system designed to handle everything from small to very large projects

**GitHub?**

GitHub is a Web-based Git version control repository hosting service. So, Git which provide revision control over code. GitHub is web-based version control. Git can be used in local project also which will carry out management of code locally only. It is not distributed nor centralized in terms of storage. But, GitHub will store git managed repositories on web. So, GitHub will contain a copy of local Git managed project so it is centralized and can be configured to use by many persons.

**How to use Git?**

To use GitHub for storage or hosting of your repositories, create a GitHub account which is free. For windows, you have to download and install Git. Git has its own command line called GitBash.

<https://docs.gitlab.com/ce/gitlab-basics/start-using-git.html>

**Three States of Code over Git**

Git has three main states that your files can reside in: **committed**, **modified**, and **staged**:

A screenshot of a cell phone

Description generated with high confidence**Committed :-**  means that the data is safely stored in your local database.

**Modified** :- means that you have changed the file but have not committed it to your database yet.

**Staged** :- means that you have marked a modified file in its current version to go into your next commit snapshot.

The basic Git workflow goes something like this:

* You modify files in your working tree. It is stored locally to current user/.git folder. First changes are made in only in workspace which can be then copies to local git folder.
* Now, we can choose the changes made by us , which adds only those changes to the staging area. It is used to keep track of changes made by current user. So, can be utilized to check in future also.
* You do a commit, which takes the files as they are in the staging area and stores that snapshot permanently to your Git directory.

<http://www.vogella.com/tutorials/EclipseGit/article.html>

**To use GitHub in Eclipse IDE**

We need install **EGit** for Git support in eclipse ide. Now, it will add new view to ide which are *Git Repositories, Git Staging, Git Tree Compare* etc. Git Repositories will provide tools for creating, adding or cloning the git repositories.

For existing project on ide, if you want to store in GitHub follow these:

1. Create GitHub repository with empty content by login to GitHub account.
2. Copy the git url.
3. Right click on project root folder -> Team -> Share Project -> select Git -> click Next.
4. Now, the current project is not a git managed project. So need to create local repository which will host the project.
5. Check first check box prompting *User or create repository in parent folder of project.* Now click on *Create Repository* -> click on Finish.
6. Local git is created. Now, we have store a copy of this in GitHub. Right click project root -> Team -> Commit.
7. This box will ask for commit message, provide one. Now select the files you wish to put in.
8. We have two options Commit & Commit and Push. Commit will store changes locally. Commit and Push will store locally and will update in GitHub repository also.
9. If you click on commit and from the same menu Team -> Push to Upstream will store in GitHub.

For staging, make some changes in existing project. Go to Team -> Stashes -> Stash changes. Give stash name. Now the changes are in staging area. To review it. Go to Team -> Synchronize workspace -> right click file to be checked and select compare with workspace. We can see both changed and unchanged version of file side-by-side. Copy the changes to local files. Now we can commit the as discussed above.