GIT AND GITLAB/GITHUB/BITBUCKET SUMMARY

Git is a version control system used for tracking changes in files. It was designed for coordinating work among programmers working on the same project. Github, gitlab and bitbucket are code repositories while git is the version system. The difference between github and bitbucket is that bitbucket has restrictions on their free tier such as the number of collaborators while github does not have such limitations.

For windows users to use git in their machine they may have to download git. By downloading git, it comes with git bash, a command line interface that can be used to issue git commands.

Let's familiarize ourselves with terms you will meet whilst using github. The term clone means is to make a copy of something that exists. So the command git clone is used to target an existing repository and make a clone of it. The command git add is used to add files to that have been changed to the stage and ready to commit the changes. After git add, we do a git commit command. This command is used to commit the changes made to the code. It's advisable to use the "-m" tag and write a descriptive message of the commit so that the other team members can understand why you did the commit and also for future reference. After committing the changes we push our changes to the git repository by using the command git push origin <<branch name>>. When we want our code to be at per with one of the branches we have to do a git pull and we specify from which branch.

During the process of pulling the latest changes, we may encounter conflicts where some lines of code from the incoming branch have differences with our lines. In this situation we do have to merge the changes. We either decide to merge the incoming changes or maintain our version.

An important feature of git that comes up a lot is the concept of branches. A branch is an independent development route, which can later be connected back to the main branch.The command git branch is used to create, list, rename and delete branches. We use branches when adding a small or big feature. When testing some logic and do not want bad code in the main branch. Also for fixing bugs and to separate releases.

The following are types of branches and their purposes:

- ➔ Master branch. This branch holds the latest version of the fully working code. This is where the current code in production resides.
- ➔ Develop branch. This branch complements the default main branch. It is the integration branch for features.
- ➔ Feature branch. This branch is used when we want to add a feature to the application.
- ➔ Release branch. This branch is used when we want to test the features we have added. The code that resides on the testing server comes from the release branch.
- ➔ HotFixes branch. This branch is used to checkout the master branch and try to fix issues detected in the main branch. Also one can use it to do experiments on the code without affecting the main branch.
- ➔ Bugfix/issue. This branch is used in case an issue is detected on the release branch. The release branch is checked out and then the issues are fixed.

The git checkout command is used to switch from one branch to another.

Git workflows

A git workflow is a recommendation for how to use git to accomplish work in an effective and efficient way. A git workflow encourages users to leverage the git effectively and consistently. This is where the different types of branches that were discussed previously come to play. A git workflow encourages the use of the different branches for different use cases. The main/master branch should only be used to track released code. Merges to the main branch comes from releases and hot fixes. We are encouraged to use the branches for their purposes.

Since git is a collaborative platform conflicts may arise. To avoid such conflicts the team should agree on a naming convention for the branches. It is important to have order in a collaborative environment.

Git tagging

Git tagging is used to mark/take note of the specific points in the repository history as being important. It is used to mark releases. Tags are pushed to the repository using the command; git push origin --tags. One can also play around with the code in the tag branch.

Github issues is a neat feature that tracks anything which is wrong with your source code. It can be found under the tab issues in the github site.

Git Gist

This is a feature that allows you to share code snippets.it is used when sharing code techniques or functions which are not complete applications. The gist is in a subdomain so to access it one has to type gist.github.com in their browser and they will be able to create gists and share their code.

Conclusion

Git is a system used for version control for source code. Github, gitlab and bitbucket are platforms to host your repositories. Git is useful in that it can make code portable in that provided you can access a computer and your github, gitlab or bitbucket account, then you can work on the code on a different machine. Github has a tab called issues that can be used to track anything wrong with your source code. Git gist is a useful feature that allows one to share code snippets easily. Git offers branches that are independent development routes that can be used to experiment and fix issues with the source code. Git is useful as if you mess up with your code you can just rollback to the previous version/commit.

References

https://developers.decoded.africa/how-to-get-started-with-git-and-github-repository-in-your-windows-machine/