Name: Vidya K N

Assignment 3

1. What is a distributed VCS?

Ans:

- Distributed Version Control System:

Every client or user can work locally and disconnected which is more convenient than centralized source control and that's why it is called distributed. We don't need to rely on the central server, we can clone the entire history or copy of the code to hard drive.

2. What are the benefits of using Version Control System?

Ans:

- The Version Control System helps manage the source code for the software team by keeping track of all the code modifications.
- It protects the source code from any unintended human error and consequences.
- The team working on the project continuously produces new source codes and keeps making amendments to the existing code. These changes are recorded for future references and can be used if ever needed in the future to discover the root cause of any particular problem.

3. What language is used in Git?

Ans:

- C is the programming language in which Git is written. C language makes Git fast by evading runtime overheads linked with other high-level programming languages.

4. Mention the various Git repository hosting functions.

Ans:

- **Github:**The largest community website for software development, and it still has some of the best tools for issue tracking, code review, continuous integration, and general code management.
- **GitLab:**GitLab is probably the leading contender when it comes to alternative code platforms. It's fully open source.
- **Bitbucket:**Bitbucket has been around for many years. In some ways, it could serve as a looking glass into the future of GitHub.
- **SourceForge:**The granddaddy of open source code repository sites is SourceForge. It used to be that if you had an open source project, SourceForge was *the* place to host our code and share releases.

5. What is a repository in Git?

Ans:

- Repositories in GIT contain a collection of files of various different versions of a Project. These files are imported from the repository into the local server of the user for further updations and modifications in the content of the file. The Version Control System is used to create these versions and store them in a specific place termed as a repository.

6. How can you create a repository in Git?

Ans:

- In the upper-right corner of any page, use the drop-down menu, and select New repository**.**
- Type a short name for your repository. For example, "vidya".
- Optionally, add a description of your repository. For example, "My name is vidya and……. "
- Choose a repository visibility.
- Select Initialize this repository with a README.
- Click Create repository.

7. What is 'staging area' or 'index' in Git?

Ans:

- These files are also referred to as "untracked files." Staging area is files that are going to be a part of the next commit, which lets git know what changes in the file are going to occur for the next commit. The repository contains all of a project's commits.

8. What is git stash?

Ans:

- git stash when you want to record the current state of the working directory and the index, but want to go back to a clean working directory. The command saves your local modifications away and reverts the working directory to match the HEAD commit.

9. What is the difference between the 'git diff 'and 'git status'?

Ans:

- **'git diff '** depicts the changes between commits, commit and working tree, etc. whereas 'git status' shows you the difference between the working directory and the index, it is helpful in understanding a git more comprehensively.
- **'git status':** 'git diff' is similar to 'git status', the only difference is that it shows the differences between various commits and also between the working directory and index.

10. What is the difference between 'git remote' and 'git clone'?

Ans:

- **Git remote:** git remote add just creates an entry in your git config that specifies a name for a particular URL. You must have an existing git repo to use this.

- **Git clone:** git clone creates a new git repository by copying an existing one located at the URI you specify.

11. What does a commit object contain?

Ans:

- The commit object contains the top-level tree object hash, parent commits hash(if any), author and committer information, commit date and commit message.

12. How to remove a file from git without removing it from your file system?

Ans:

1. Add all the files, individually or in a folder, that you want to remove from the repo but keep locally to **.gitignore**.
2. Execute **git rm --cached put/here/your/file.ext** for each file or **git rm --cached folder/*** if they are in a folder. (It is /* because you need to escape the *)
3. Commit your changes.
4. Push to remote.

13. Can you explain the Gitflow workflow?

Ans:

- The Gitflow Workflow defines a strict branching model designed around the project release. This workflow doesn't add any new concepts or commands beyond what's required for the Feature Branch Workflow. Instead, it assigns very specific roles to different branches and defines how and when they should interact.

14. What are the different ways you can refer to a commit?

Ans:

In Git each commit has a unique hash. These hashes are used to identify the corresponding commits in various scenarios, for example, while trying to checkout a particular state of the code using the git checkout {hash} command.

Along with this, Git maintains a number of aliases to certain commits, known as refs. Also, every tag that is created in the repository effectively becomes a ref and that is exactly why you can use tags instead of committing hashes in various git commands. Git also maintains a number of special aliases that are changed based on the state of the repository, such as HEAD, FETCH_HEAD, MERGE_HEAD, etc.

In Git, commits are allowed to be referred to as relative to one another. In the case of merge commits, where the commit has two parents, ^ can be used to select one of the two parents, for example, HEAD^2 can be used to follow the second parent.

And finally, refspecs are used to map local and remote branches together. However, these can also be used to refer to commits that reside on remote branches allowing one to control and manipulate them from a local git environment.

15. Explain the difference between reverting and resetting.

Ans:

- **Resetting**:

git reset is used when we want to unstage a file and bring our changes back to the working directory. git reset can also be used to remove commits from the local repository.

- **Reverting**:

git revert is used to remove the commits from the remote repository. Since now our changes are in the working directory, let’s add those changes to the staging area and commit them.