|  |  |  |  |
| --- | --- | --- | --- |
| **Subject: -** DevOps(ITL503) | | | |
| **Year: -**TE IT | **Class: -**1 | | **Batch: -** I2 |
| **Roll No.: -** 43 | **Student Name:** - Akash Bhairavnath Jagdale | | |
| **Date of Experiment: -** 26/08/2022 | | **Date of Submission: -** 26/08/2022 | |

**Experiment No. 3:**

**GIT Remote and Local Repositories**

**Title: -** Git and GitHub

**Aim: -** Create a GitHub account and perform various GIT operations on Remote Repositories using GIT Cheat – Sheet. Study and Implement Advance version control commands for GIT and GitHub.

**Theory: -**

**What is remote repository?**

Remote repositories are versions of your project that are hosted on the Internet or network somewhere. You can have several of them, each of which generally is either read-only or read/write for you.

**GIT Hub: -**

GitHub is a remote repository hosting service for git version control system. It provides free hosting services with pubic and private repositories and many other features through premium. It has many exclusive features like fork, pull request etc. Github is probably most famous remote hosting service and acts crucial role in team work.

**Difference between GIT and GitHub: -**

|  |  |
| --- | --- |
| **Git** | **GitHub** |
| It is a version control system | It provides services related to version control system git. |
| Creates and manages repositories on local machine | Manages git repositories on cloud |
| Used only for managing history and versions of source code | Used for sharing different repositories and managing them |

**Difference between GIT fork and clone: -**

|  |  |
| --- | --- |
| **Fork** | **Clone** |
| It is an exclusive github feature | It is exclusive git feature |
| It is used to copy entire repositories on github. | It is used to copy entire repositories on local computer |
| The changes on original repository can be merged and with forked repository through pull request | Changes made on local repository can be merged directly to remote repository |

**GIT stash and pop: -**

**Git stash:**

In scenario where we have made changes in repository in a branch but we had to do them in another branch then we can transfer all this changes in current branch to a temporary index files named stash files. Git stash used to do this.

**Git pop:**

If any stash files are present in the repository then this changes can be moved to any intended branch using git pop command. After transferring this changes into current branch we can commit them where we originally wanted.

**GIT Rebase: -**

If we had made a branch in our git repository right after commit ‘C1’ and we had made some commits named ‘F1’ ‘F2’in this separate branch. We then make another commit C2 in master. If we have to make this git history more simpler after merge then we can use rebase command. This command sets base of another branch to recent commit in master branch.

**Difference between GIT Rebase and Merge: -**

|  |  |
| --- | --- |
| **Rebase** | **Merge** |
| Combines two branches directly without adding new commit. | Combines two branches into one by adding a new commit between master and current branch |
| Base of merged branched is pointed by recent commit | Base of merged branched is pointed by merge commit |

**GIT Commands: -**

|  |  |
| --- | --- |
| **Commands** | **Explanation** |
| Git clone | Copies entire repository from cloud. Any changes in copied can be pushed to remote repository or its changes can be pulled in copied repo. |
| Git fetch | Checks whether there are changes in remote repository |
| Git push | Pushes changes from local repository to remote repository. |
| Git pull | Pulls or downloads any changes from remote repository to local repository |
| Git rebase | Merges two branches directly through recent commit in master branch |
| Git stash | To move changes from one branch into temporary files called as stash files |
| Git stash pop | To move stash files changes into current branch |

**Advantages of using GitHub: -**

1. GitHub provides free hosting services.
2. It is currently best remote service provider for Git
3. It has many of its own plugins to manage source code and integrate it with other tools.

**Implementation: -**

Create a GitHub account.

**Part1:**

1. Push the MyDirectory folder (created in exp.2) to the remote repository.

2. Add another file called “File4.txt” and add a few lines to it.

3. Make a commit on a remote repository.

4. Make changes to a copied file (i.e file 1 from exp2) on a remote repository and make a commit. 5. Pull to the local repository.

6. Display one-line log.

7. Create a clone of the repository.

8. Fork public repository.

9. Copy forked public repository to your local system.

**Part2: (on local repository)**

1. Go to the newly created branch in your “MyDirectory” project (the one we started in exp.2) 2. Add few new lines to existing file (eg.file 1)

(same file exists on main branch, mistakenly you made changes in branch’s (file1) file so...) 3. Save file in stash.

4. Go to the main branch (master) and pop the changed contents to file (file1 of master) in the main branch. 5. Make a commit.

6. Display one-line log.

**Part3: (on local display difference between rebase and merge)**

7. Perform rebase operations for branches you have created. (if branches have already merged, create a new branch). 8. Display one-line log and one-line graph.

9. Create another branch, add file modifications. Merge branches and Display one-line log and one-line graph.

**Conclusion: -** Hence, we successfully pushed and pulled files on our GitHub remote repository.