**Version control system – Git**

* **Basic local operations : -**

*1] Create a new local git repository on to your machine and configure it for*

*user.name and user.email attribute.*

$ git config --global user.name "Rhishabh"

$ git config --global user.email [rhishabh\_hattarki@persistent.com](mailto:rhishabh_hattarki@persistent.com)

$ git init

*2] Create below files in directory– File1.txt, IgnoreFile.txt and FileToDelete.txt, now start tracking only the File1.txt and FileToDelete.txt, and ignore the IgnoreFile.txt by using .gitignore.*

$ touch File1.txt

$ touch FileToDelete.txt

$ touch IgnoreFile.txt

$ touch .gitignore

$ echo “IgnoreFile.txt” >> .gitignore

*3] Commit these changes to repository.*

*Use commit message as “This is First Commit, Tracking File1 and FileToDelete”*

*Note: - only File1.txt and FileToDelete.txt should be tracked and not the IgnoreFile.txt*

$ git add File1.txt FileToDelete.txt

$ git commit -m “This is First Commit, Tracking File1 and FileToDelete”

*4] Make 2 more commits by making following changes in the File1.txt*

*Add changes to file as:-*

*Change1 – commit with message as “Change1 in File1”*

*Change2 – commit with message as “Change2 in File1”*

*And if you check the log it should display in following manner:-*

*Change2 in File1*

*Change1 in File1*

*This is First Commit, Tracking File1 and FileToDelete*

$ echo “Change 1” >> File1.txt

$ git add File1.txt

$ git commit -m “Change1 in File1”

$ echo “Change 2” >> File1.txt

$ git add File1.txt

$ git commit -m “Change2 in File1”

*5] Delete the FileToDelete.txt file from directory and commit with message as “FileToDelete is deleted”*

$ rm -rf FileToDelete.txt

$ git rm FileToDelete.txt

$ git commit -m “FileToDelete is deleted”

*6] Display logs with below filter condition:-*

*- Show all the commit logs.*

*- Show last three logs in one line*

*- Show only commits that occur between from date and to date*

*- Show only those commit message that contains “File1” as in commits.*

(Assuming from, not inclusive -> 3/2/2021 and to, inclusive -> 4/2/2021)

$ git log

$ git log -3 --oneline

$ git log –since “3/2/2021” –until “4/2/2021”

$ git log –grep=”File1”

*7] Permanently undo a “FileToDelete is deleted” commit snapshot (performed in step 5).*

(Assuming FileToDelete commit was the previous version, use sha1 hash/tag instead)

$ git reset –hard HEAD~1

*8] Rename File1.txt to TestFile.txt and commit it*

$ mv File1.txt TestFile.txt

$ git rm File1.txt

$ git add TestFile.txt

$ git commit -m “Renamed File1.txt to TestFile.txt”

*9] undo change made in above step 8 but retain the history log*

(Assuming above change is in previous version, use sha1 hash/tag instead)

$ git revert HEAD~1