GIT – GLOBAL INFORMATION TRACKER

* Git is used to track the files.
* It will maintain multiple versions of the same file.
* It is platform-independent.
* It is free and open-source.
* They can handle larger projects efficiently.
* They save time and developers can fetch and create pull requests without switching.
* It is 3rd generation of VCS (version control system).

MANDATORY STEPS TO DO INSTALL GIT :-

* yum install git -y 🡪 To install git
* git init . 🡪 We need to create empty .git repository – inside root/.git folder created
* ls -al 🡪 To get hidden files. (here we get .git folder)
* git --version 🡪 To check git installed or not.

Note :- Without .git folder impossible track files. So, we need to initial .git folder.

|  |  |
| --- | --- |
| *GIT COMMANDS* |  |
| touch filename | To create a file. |
| git add filename   * git add \* * git add . | Now add that created file to git – for tracking.   * For all files. * For all files including hidden files. |
| git status | To check the file tracking or not. |
| git commit -m “commit msg u want” filename /\*/.   * m - message | To commit the file with message – saved |
| git rm –cached filename | To untrack |
| git log | To check history |
| git log -2 | To see top 2 commit |
| git log --oneline | To see only commit id's and msgs |
| git show commit\_ID filename | To see details of commit filename & details. |
| git log --follow --all filename | To see all commits for a single file |
| git revert |  |

GIT-CONFIG

|  |  |
| --- | --- |
| git config user.name “naveen\_silver” | To config user name |
| git config user.email “naveensilver136@gmail.com” | To config user email ID |

GIT – IGNORE ?

GIT – DELETE

GIT - BRANCH

* Multiple developers can work on a single project.
* We can maintain the versions od the application.

|  |  |
| --- | --- |
| *BRANCH COMMANDS* |  |
| git branch | To get list of all branches / default branch  default branch : master |
| git branch branch-name | To create a branch  ex:- git branch devops |
| git checkout <shifting branch-name> | To shift one branch to another branch  ex:- git checkout devops {master – devops} |
| git checkout -b <new branch-name> | To create a branch as well as shift to new branch  ex:- git checkout -b aws |
| git branch -d/D <deleting branch-name> | To delete a branch  ex:- git branch -d devops  [when you’re deleting a branch , you shouldn’t present in deleting branch ] |
| git merge <branch-name> | To merge one branch files to another branch  ex:- git merge aws {master 🡨 aws (aws files merge)} |
| git cherry-pick <paste commit\_Id> | To merge a particular branch file  ex:- first copy the particular file commit\_id{aws branch file commit id and shift the branch master } |
|  |  |

Merge conflicts:-

Def ?