**GIT BASIC COMMANDS**

**Install & Configure the tool For the first time:**

$ sudo apt-get install git

To install git on linux machine.

Configure user information for all local repositories

$ git config --global user.name "[name]"

Sets the name for your commit transactions.

$ git config --global user.email "[email address]"

Sets the email address for your commit transactions.

$ git config --list

To check your configuration settings.

**HELP**

$ git help <verb>

$ git <verb> --help

To get the usage of a command.

$ man git

To get the complete manual page.

**CREATE REPOSITORIES**

Start a new repository (or) obtain one from an existing URL.

$ git init

To initialize an existing directory as a Git repository

$ git clone [path-to-remote-repository]

To retrieve an entire repository from a hosted location via URL.

**ADD & MOVE**

$ git add [file]

To add changes in the working directory to the staging area.

$ git mv file\_from file\_to

To rename the file in the working directory.

**COMPARISION & COMMIT**

Reviewing and commiting the changes made.

$ git status

Lists all new or modified files to be committed.

$ git diff

Shows file differences not yet staged.

$ git diff --staged

Shows file differences between staging and the last file version.

$ git commit -m "[descriptive message]"

Records file modified data permanently in version history.

$git log

show all commits in the current branch’s history.

**BRANCH & MERGE**

Isolating work in branches, changing context, and integrating changes.

$ git branch

list your branches. a \* will appear next to the currently active branch.

$ git branch [branch-name]

create a new branch at the current commit.

$git checkout [branch-name]

switch to another branch and check it out into your working directory.

$git merge [branch]

merge the specified branch’s history into the current one.

**SYNCHRONIZE CHANGES**

$ git fetch [remote]

To fetch any new work that has been pushed to that server since you cloned it.

$ git pull

To automatically fetch and then merge that remote branch into your current branch.

$ git push [remote] [branch]

To push to remote server.

**TAGGING**

$ git tag <tag\_name> <commit ID> -m "tag\_message"

To tag a commit ID.

$ git tag

Shows the list of available tags.

$git push --tags

To publish your tags.

**Use case:**

In a remote server we have the git repository, which has the source code. The source code can be accessed by many teams and multiple tasks can be accomplished in parallel. Each member of a team can clone the complete repository to their local machine and can modify or add the code, review and commit it by creating a branch. The branch should be merged with the original default branch, pull to get any updates since the last clone and push it to server.

**Example:**

step1: clone the remote git repositiry

$ git clone [path-to-git-repository]

step2: check for the list of available branches

$ git branch

step 3: create a new branch

$ git branch new

step 4: swith to the new branch from the

$ git checkout new

step 5: Do some changes

$ vim sample.txt

(Add some data)

step 6: To know the status of the new or modified file

$ git status

step 7: To know the lines that are added or deleted

$ git diff

step 8: To add the modified file

$ git add .

step 9:

$ git staus

step 10: To commit the changes

$git commit -m “commit\_message”

step 11: To see the commit history in the branch

$git log

step 12: switching back to master branch

$git checkout master

step 13: to merge the branch “new” to branch “master”

$git merge new

step 14:

$ git log

step 15: To delete the branch

git branch -d new

step 16: check whether the source code is updated from the time you have cloned.

$git pull

step 17: Apply the changes to remote server.

$git push origin master