It is a continuous integration and configuration management tool.

1. GIT 🡪 local repository (configuration management tool)
2. GITHUB 🡪 remote repository
3. Continuous integration

**Git and GitHub:**

**Pre-requirements:**

* You need an account with GitHub
* Create a new project/repository in GitHub

**Step 1:** <https://github.com/> 🡪 create new Account in GitHub application.

**Step 2:** Create new repository in GitHub 🡪 click new repository 🡪 give project name 🡪 select public or privet 🡪 click create repository.

After clicking the create repository it will generate repository URL like: (copy the url after creating the repository)<https://github.com/training/inetBankingV4.git>

**Step 3:** Install git plugin in eclipse. Go to eclipse 🡪 click help menu 🡪 click eclipse market place 🡪 search git 🡪 press go button 🡪 select EGit – Git integration for Eclipse 4.6.0 🡪 click install button.

**Git installation:**

To download the latest version of Git, click on the link: <https://git-scm.com/download/win/>

**GitHub Remote repository url:**

<https://github.com/training/inetBankingV4.git> Download Git applications and install them in to our machine.

After installation create one folder on local device and give name of the folder is inetBanking and make it as a local Git repository.

**Creating repository:**

Go to the folder 🡪 right click on inside the folder 🡪 select Git bash here 🡪 after selecting the Git bash command line was open automatically.

This command line is used to make a directory as a local repository.

Initialing the Git command 🡪 git init

After giving the initialing command it will show **initialized empty Git repository in C:/InetBAnking/.git/**

Clear command is used to clear the repository.

**Establishing connection Git and GitHub Connection:**

Making Remote repository command 🡪 git remote add origin “<https://github.com/pavanoltraining/inetBankingV4.git>”

**create/clone git repository in eclipse:**

Go to eclipse 🡪 go to quick access option 🡪 search git repository 🡪 it shows some git repository’s 🡪 select clone a Git repository and add the clone to this view option on top right corner of small window 🡪 click 🡪 enter the repository url (<https://github.com/training/inetBankingV4.git>) 🡪 click next 🡪 again click next 🡪 click finish. Now our project is ready to work in local repository. Local repository is ready but our code is not a part of local repository.

**Commit project to Git and GitHub repository:**

Go to eclipse 🡪 right click project 🡪 select team 🡪 select share project 🡪 select the cloned repository 🡪 finish.

**Every time do the same whoever you do changes:** select your project 🡪 right click 🡪 team 🡪 commit.

**Pushing the code Git to GitHub (local to remote):**

select your project 🡪 right click 🡪 team 🡪 commit 🡪 select all the file and drag them upstaged changes to staged changes 🡪 give commit message like pushing InetBanking to Git repository 🡪 click push and commit button 🡪 click next 🡪 enter user name and password of your created on sign up process(email id password) because Git is connected to GitHub repository 🡪 click ok.

**Pulling the code or file GitHub to Git (remote to local):**

select your project 🡪 right click 🡪 team 🡪 click pull option 🡪 click ok.

**Pull the files from remote repository to local Git repository:**

git pull origin master 🡪 this command is used to pull the file from remote repository to local repository. Whatever files we have under master branch in the git hub all the files will be part of a local repository by using pull command.

**Work with repository by using Command line:**

**Pull the documents into local Git repository:**

create one file or folder in to the local repository directory

command 🡪 git status 🡪 this command is used to retrieve the how many files present under the Git repository

here we need to pull the document local to remote repository. but, File or folder we cannot directly move into the Git repository. We need some intermediate layer called index. Before we adding the file to the Git repository we need to add file into the index. Then from index to git repository.

Add commit

File(Test1.txt)

Git repository

Index

**Adding file to index before committing:**

Command 🡪 git add test1.txt 🡪 this command is used to add the file in to the index. After the command the file is ready to commit git repository.

**Pushing single file using commit commands:**

Command 🡪 git commit –m “committing my first file into local repository” 🡪 which is used to move the file in git repository. After committing the file which is used to perform we can push the same file in to the GitHub repository.

**Pushing multiple file using commands:**

Create multiple files in our local directory called inetBanking 🡪 git status 🡪 which is used to check the status of file.

Command 🡪 git add –A 🡪 this command is used to add all the files into the index.

Command 🡪 git commit –a –m “committing all files into git local repository”🡪 this command is used to commit all the files into the local repository.

Command 🡪 git log 🡪 this command is used to retrieve the date time everything.

**Git to GitHub (local repository to remote repository):**

**Pulling the Files:**

pull command is fetches changes from a remote repository to a local repository. It merges upstream changes in your local repository, which is a common task in Git based collections. But, first you need to set your central repository as origin using the command.

Command 🡪 git pull origin master

Press ctrl+x to exit the pull window

**Pushing the file:**

Push command is used upload the files local repository to remote repository

command 🡪 git push origin master.

**Continuous Integration:**

**Phase -1: implementation:**

1. Create maven project
2. Update pom.xml
3. Create folder structure
4. Create page objects
5. Create basic test case
6. Add logs to test case
7. Read common values from properties file
8. Run test case on desired browser
9. Add extend report
10. Create data driven test case
11. Adding new test cases

**Phase -2: Execution:**

1. Run test case with maven pom.xml
2. Run test cases through maven CLI (Command Line Interface)
3. Run test cases using run.bat
4. Run test cases using Jenkins (using bat file)

Run test case with maven pom.xml:

Plug-in 🡪 used to run your maven project.

Maven-compiler-plugin

Maven-surefire-plug-in

Run test cases through Command Line Interface (without using eclipse):

[Maven – Download Apache Maven](https://maven.apache.org/download.cgi?.)

After download the source file we have to configure the path go to system properties 🡪 click advanced settings 🡪 click environment variables 🡪 inside the system variables select path variable 🡪 click edit 🡪 click new paste the maven bin file location (C:\Users\keerthiraja.sp\Downloads\SeleniumJarFiles\apache-maven-3.9.6-bin\bin) 🡪 click ok.

After completing configuration use command prompt to execute the maven project

Command 🡪 mvn –version