Create Account in GitHub: -

GitHub Credentials: -

Username = prathibhasuresh50@gmail.com

Password = Githu@123

Install Git in Local machine.

After Completing your framework for the first time.

Go to Github 🡺 Create a new repository 🡪 copy urI 🡪

Go to framework directory and type cmd 🡪 in the CLI

1)Initialize git

* Git init

2)Establish connection between server and client

**🡺** git remote add origin “Copied repo URL”

3)Check the status before committing

🡺git status

4)Add all into staging to commit

🡺 git add \* / git add –A 🡪 Now everything is in stage

5)commit the code

🡺git commit –m “any comment”

6)To Push the code we need to connect remote repository to our local directory

🡺git remote add origin “Repo URI”

7)Push the code to remote repo(github)

🡺git push origin master

Now we code is in remote repo.

8)Clone code from GitHub to a new directory to update or work on it🡪 go to project directory and type cmd🡪 In the CLI

🡪git clone “repo uri”

9)Update the code and try to commit

🡪git status

🡪git add \*

🡪git status

10) commit the changes which are in stage

🡪git commit –m “Changes after clone project”

11)push the changes (for the second time no to establish the connection to remote rope)

🡪git push origin master

Now all changes are pushed to GitHub from the new directory

Go back to old directory and see the new pushed changes. Here no need to clone for the second time we need to use pull command.

12) Go to old directory and pull the new changes

🡪git pull origin master

Branching in GIT: (go to old directory)

1)To create a new branch

🡪git checkout –b “name of the sub branch”

2) Check you are in which branch

🡪 git branch

The make the change in devlop branch and commit and push

🡪git status

🡪git add \*

🡪git status

🡪git commit –m “comments in the sub branch changes”

🡪git push origin devlop

Then go to new directory and try to switch branch and pull the latest changes

🡪git checkout devlop

🡪git pull origin devlop

After changes in devlop old directory need to merge to master branch

🡪git checkout master

🡪git pull origin master

🡪git merge devlop 🡪 merge to active branch active branch is here is master

**Jenkins Pipeline:**

A pipeline is a collection of jobs that brings that software from version control into hands of the end user by using automation tools.

It is a workflow used to achieve continuous delivery in software development process.

Pipeline can be implemented with below 2 methods:

1. Jenkins Build Pipeline Plugin (Suitable for small scale Applications).
2. Jenkins Pipeline Projects (Large Scale Application with many jobs)

In second Approach Pipeline is created by executing the Jenkins file.

A Jenkins file is text file that the entire workflow as code.

The Jenkins is written using the Groovy DSL(Domain Specific language).

It is written based on two syntaxes, namely

Declarative Pipeline Syntax.

Scripted pipeline Syntax.

Using Jenkins Build Pipeline plugin:

Build - mvn install

Deploy – Shell

Testing – mvn Test

Release – tasks

Using Jenkins Pipeline projects:

Build

Unit Testing Coverage – Tools called Sonar Cube

Deploy into Dev

Smoke Testing

Integration Testing

Unit Testing

Deploy into QA

End to End Testing

API Testing

Quality Check

Release into Prod.

We will write one script