

Day 1 :

26/08/2023

SDLC : Software Development Life cycle

Water fall model

Requirement gathering 6 months

Plan

Design

Coding / testing

Build the project development mode

Deploy the project production mode

Provide the service

Increment model

V model

Agile model

Sprint : time duration to develop small module 1 or 2 week. Login page, feedback page

Dashboard

DevOps : Development and Operation

Development people develop the application using any language like java, python, etc

Operation team interact with customer or end user client and they are responsible to maintain the application.

Admin

Developer

Tester

Admin

Database designer

Architecture etc

Devops tools

Git and git hub

Maven and Gradle (Java development)

CI and CD tool using Jenkin (Continuous Integration and Continuous deployment or delivery)

Selenium tool (testing)

Ansible tool (monitoring tool as well as configuration tool)

Docker container

Nagios tool

Kubernetes

Git and github

Sub version control which help to **record** the application flow.

Dev1	login page	
Dev2	java or python code	merge the code
Dev3	database	

Unix commands

ls : this command display all files and folder present in current directory

pwd : print working directory : it show current path of terminal or command prompt.

mkdir foldername: this command is use to create the folder

cd foldername : move inside a folder

cd .. : come outside a folder or move to parent directory of current folder

rmdir foldername : to remove folder

touch filename : to create empty file

vi filename : open the file in vi mode

once open hit i key to move inside a vi editor mode

write the contents

esc : to come out from editor mode to normal mode

:wq : write and q quite (save and exit)

cat filename : it is use to read contents from a file

cat means concatenate

Git : Git is version control system which help to track or record changes done in the application or project or app.

Git also known a distributed sub version control.

First create the folder

Then create the file and write the contents

git --version

git init it is use to make local folder as git repository

init command create .git folder insider that current directory

ls -a : it display all files and folder with hidden folder.

git status : this command is use to check the current status of your repository

git add filename : to add files or folder normal local folder to tagging area.

Or

git add . : this command is use to add all files and folder present in

Current directory.

git commit -m "message" : this command use to pass the task from tagging area to

local repository.

```
git config --global user.email "akash300383@gmail.com"
```

```
git config --global user.name "akash"
```

steps

1. Create folder with any name ie Demo
2. Then create the file with any name ie test and write the contents inside that file.
3. Then open the terminal inside that folder please use `pwd`
4. `git init`
5. `git status`
6. `git add .`
7. `git status`
8. `git commit -m "message"`
9. first time we need to set config details as email and name
10. `git config --global user.email "akash300383@gmail.com"`
11. `git config --global user.name "akash"`
12. Then please commit using command as `git commit -m "done"`
13. `git status`

`git branch` : branch is like a pointer which holds more than one commit details.

By default git provides default branch. Default branch name may be `master` or `main`.

If we want to check branch details present in local repository

`git branch`

command to create user defined branch

`git branch branchname` this command is used to create user defined branch

`git checkout branchname` this command is used to switch from one branch
to another branch.

Current branch is master or any other branch

`git merge branchname` this command adds all tasks in current branch

`git branch -D branchname` this command is use to remove the branch

Demo.java

int a; akash branch

int b; Vikash branch

Remote repository help us to share the code between two or more than one tabme.

Git hub

Git lab

Bitbucket

Aws

Azure

Private cloud etc

git hub : it is a type of remote repository provided by micro soft organization.

Day 2 :

27/08/2023

We want to connect local repository with remote repository

1. Token base authentication
 2. SSH Client
- More

To connect local repository with remote repository

`git remote add origin URL`

`git remote add origin https://github.com/Kaleakash/test_rep.git`

`git remote add origin https://token@github.com/Kaleakash/test_rep.git`

`git push -u origin main` (it is use to push the code)

how to resolve the conflict

1. first create Repo2 folder
2. then open terminal inside that folder
3. create sample file
4. add some data 1st, 2nd
5. using git init make folder as repository
6. git add .
7. git commit -m "done changes in master branch"
8. create the branch
9. git branch akash
10. git checkout akash
11. in akash branch we will add 3rd and 4th message.
12. Then git add .
13. Then git commit -m "in akash branch done some changes in sample file"
14. Create another branch with name as Vikash

15. `git checkout -b Vikash` (it will create the branch and switch to that branch)
16. in sample file in Vikash we will add the message as 5th and 6th.
17. Then `git add .`
18. `Git commit -m "done change in sample file by Vikash branch"`
19. Please move the master branch ie `git checkout master`.
20. Please verify current branch using command as `git branch`
21. Then in master branch merge the code from akash branch
22. `Git merge akash`
23. Using `cat sample` read the data from sample file
24. Out must be 1st, 2nd, 3rd, 4th
- 25.

Download or clone the repository

1. Create the folder with any name `ie devopstrainig` In VM
2. Then open the terminal
3. `git clone URL`
4. `git clone`
https://github.com/Kaleakash/devops_aug_2023_trainig_batch.git
5. use `ls` command to see downloaded folder
6. using `cd` command please move inside that folder.
7. `cd devops_aug_2023_trainig_batch`
8. `ls` command to see the more than one file.

if we do any changes in local repository
we need to add, commit and push

```
git add .  
git commit -m "done some changes in file"  
git push -u origin main
```

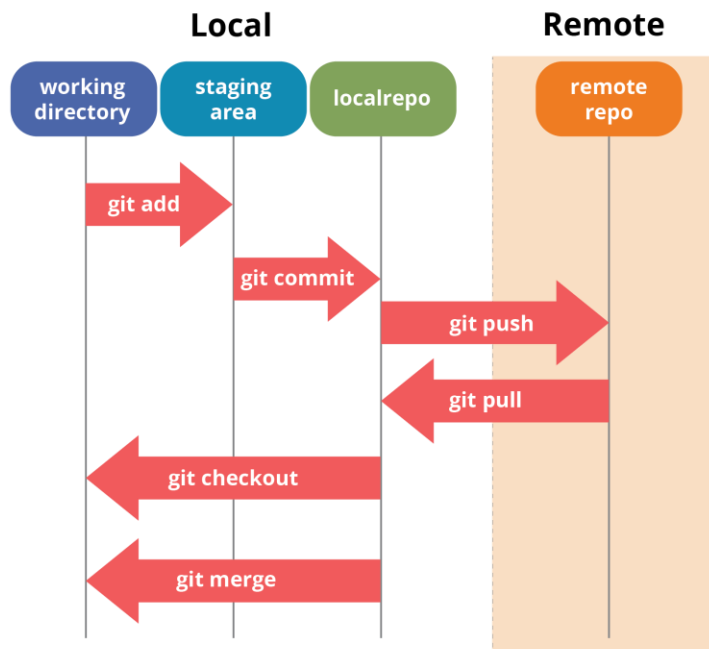
1. open the terminal inside a repository folder
2. make sure `.git` folder present using command verify `ls -a`
3. `git pull`

`git clone URL` : it help to download fresh repository in local machine

`git pull` : it will pull new updated from remote to local repository

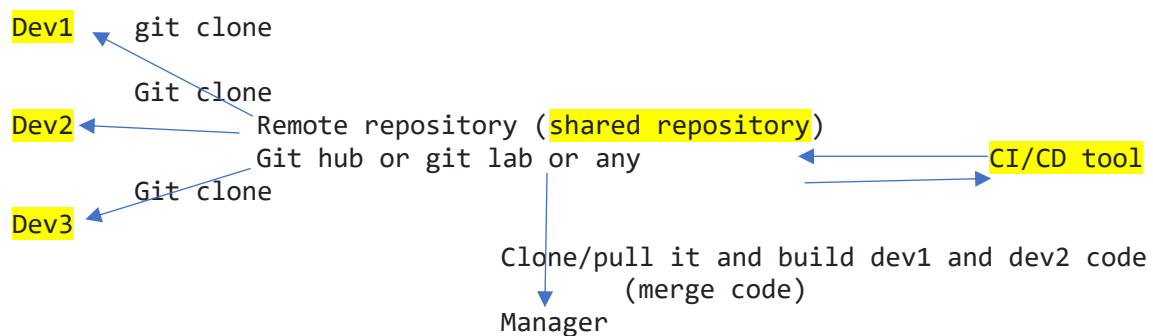
Git Vs Git hub

Git is Remote repository provided by micro soft.
 Command line or GUI tool which
 Help to interact with any remote repository



CI and CD tools

Continuous Integration and Continuous delivery or deployment



java	Git hub	Jenkin
python	Bit bucket	GoCD
angular		

Generally shared repository can be private or public.
 While creating or after created that repository we will send the invitation
 To all developer to join that repository.

Default branch can be master or main.

Dev1 need to create login page using html and css

Dev2 need to create java or python code

Dev3 need to create database tables.

If they do all their task in main or master branch after changes done they can add, commit and push the code to shared repository.

Don't do your task in main or master branch. While doing your task please create user defined branch and push that branch in remote repository.

In remote repository we will check user defined branch if code is correct we will merge that code into master or main branch.

Build phase :

You need to compile and run the application using that language.

Javac

Java java

Py python

Ng angular

Dev1 push the code in remote repository

In remote repository we need to verify the code and merge the code. And we need build the application.

After dev2 or dev3 code merge in master or main branch we can build successfully or it generate some error.

CI and CD tools.

Jenkin : it is a type of CI and CD tools. It is an open source ci/cd tool base upon java technologies. Plugin base ci and cd tool. GUI based tool.

Open the browser

<http://localhost:8080>

it will ask login details

username : admin

password : admin

in Jenkin we need to create the job. Every job responsible to build the project.

Day 3 :

02/09/2023

Open the terminal in VM.

git clone URL

next

git pull (but make sure terminal open inside that folder).

Web Service : Giving the service for web application when both the application running using different technologies

API : Application Programming interface.

Web Service

Java

python

XML/JSON

eXtensible markup language

JSON : JavaScript object notation

HDFC

XML/JSON

HSBC

Micro service :

Please refer the website the set the trigger time using crons

<https://crontab.guru/>

Day 4 :

03/09/2023

Open the terminal and start /stop jenkins service

`sudo service jenkins stop` unix

`sudo service jenkins start` start

Jenkins provide few pre installed tools like Maven, Gradle, Git, ant etc. These tools help us to run java technologies.

Jenkins Pipeline a pipeline is a collection of event or job which interconnected with each other to perform a specific task.

Maven goal

Clean

Compile

Install jar or war unit test

Test

Package

```
pipeline {  
  agent any  
  stages {  
    stage('Hello') {  
      steps {  
        echo 'Hello World'  
        sh 'git --version'  
        sh 'v'  
      }  
    }  
  }  
}
```

Maven : Maven is open source build tool we use in java technologies to build the project.

Maven goal

Clean : clean complete project doesn't matter it contains one file or more than one file

Compile

Install : install some dependencies

Test : test the project

Package : creating jar or war file

```

pipeline {
    agent any

    tools {
        // Install the Maven version configured as "M3" and add it to the path.
        maven "M3"
    }

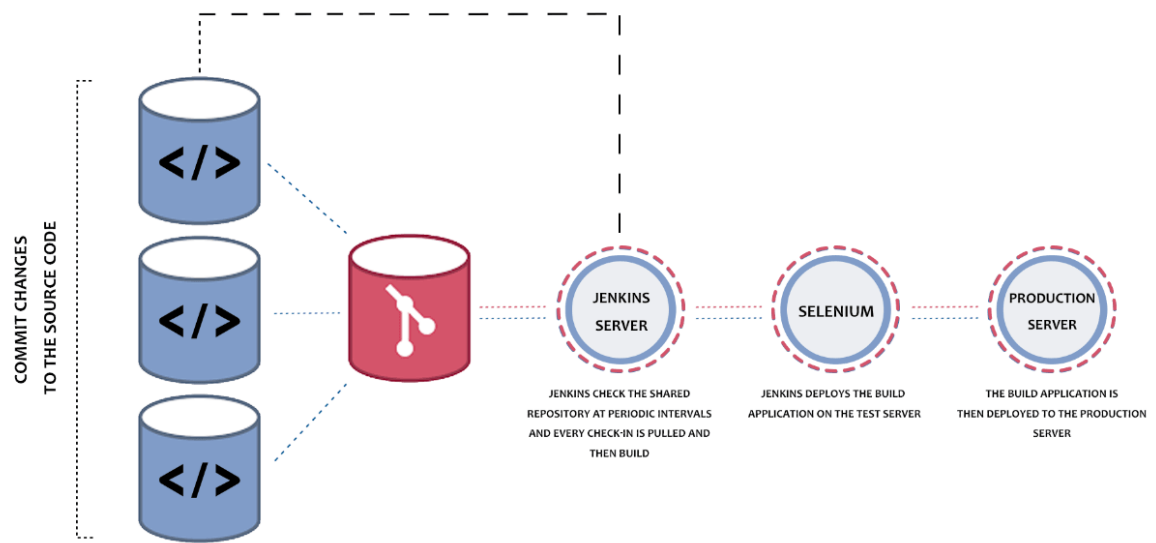
    stages {
        stage('Build') {
            steps {
                // Get some code from a GitHub repository
                git 'https://github.com/jglick/simple-maven-project-with-tests.git'

                // Run Maven on a Unix agent.
                sh "mvn -Dmaven.test.failure.ignore=true clean package"

                // To run Maven on a Windows agent, use
                //bat "mvn -Dmaven.test.failure.ignore=true clean package"
            }
        }

        post {
            // If Maven was able to run the tests, even if some of the test
            // failed, record the test results and archive the jar file.
            success {
                junit '**/target/surefire-reports/TEST-*.xml'
                archiveArtifacts 'target/*.jar'
            }
        }
    }
}

```



If we want to run more than one command with help of normal job.