**GIT & GIT HUB :**

What is the solution when :

- A developer has to manage or track changes to the code.

- When multiple developers have to work simutenously and contribute to final code

- When we have to track who or when changes were made.

- When we have to manage all the older versions of the code.

The Solution is: SOURCE CODE MANAGEMENT SYSTEM or VERSION CONTROL SYSTEM

What is a Version Control System:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

- A system that documents changes made to a file or set of files.

- manages multiple revisions to files

- manages files from multiple users.

"It is snapshot of your project overtime"

Types of Version Control System:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Local Version Control

- One to One interation between developer and the local Version System.

Challenge:

This system doesnot support managing of data when multiple developers are working parallely on a project

Eg: RCS

2. Centralized Version Control System

- In this system multiple developers are working together and can move and track there chnages on to a Central repository.

eg: CVS,subversion, perforce, TFS(team foundation Version Control) etc

3. Distributed Version Control System

This system is Distributed in nature where in all the developers will have a working Directory, local repository and a Central repository

eg: GIT, Mercurial, BITKeeper, Darcas, Bazzar etc.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What is GIT?

- its open source and is developed by Linus Torvalds.

- Distributed version Control System

- Tool that tracks changes in your code/files over time.

- Working Directory, Stagging area, local repository and remote repository

- GIT stores the data in sanpshots that is the reson that its performance is very fast and it can store huge amount of data.

What is GIT hub?

- GitHub is the website where we host all of our git repositories.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Connect to Ssh client

# ssh -i "pemkey" ussername@public ip adress

# ssh -i "dockertoday.pem" ec2-user@publicipaddress

**JENKINS:**

CI - Continuous Integration

1.Jenkins - 80% - CloudBees Jenkins

2.Circle CI

3.Teamcity

4.Bambooy

5. GitLab

6.Travis CI

7. CodeSHip

Jenkins

+++++++++++++

Jenkins is a CI server which manages and control process such as

Plan, Code, Build test, deploy operates and monitor in DevOps Env.

Jenkins:

+++++

3 Pre-Requists

1. Java - 1.8 version

2. Git

3. Maven - Build Tool ( Ant, Gradle)

#(root directory)

8080 - default

what is jenkins job?

====================

A Jenkins job is a sequential set of tasks that a user defines.

For example, a job can fetch source code from version control, compile the code, run unit tests, and more. Note that in Jenkins, the term “job” is synonymous with “project”

2 Types:

1. Freestyle job - Manual - UI

2. Pipeline - Groovy Scripting - CI/CD

what is Freestle Job?

================

This is the central feature of Jenkins.

Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

what is pipeline?

================

Orchestrates long-running activities that can span multiple build agents.

Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Jobs Creation:

Build Tool of Maven

=================

1. compile :

===========

convert into machine understandable language

2. Code Reviiew

================

reporting task, code analysis report - PMD Tool

3. Unit Test

============

execution of test cases, test report( pass or fail)

4. Code Coverage

===========

% code covered by the test executes, coverage report

5. Package

==============

its convert into .jar, .war, .ear

Build Life Cycle

=================

validate - validate the project is correct and all necessary information is available

compile - compile the source code of the project

test - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed

package - take the compiled code and package it in its distributable format, such as a JAR.

verify - run any checks on results of integration tests to ensure quality criteria are met

install - install the package into the local repository, for use as a dependency in other projects locally

deploy - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.

why Jenkins is so popular

==========================

open source

Good Plugin support

Good community support

Fast and reliable

Good OS Support

Scripted Build

Features of Jenkins

===============

Easy installation process

provides advance security

upgrades are easily available

light weight container support

Distributed Team Management

Master & Slave configuration:

++++++++++++++++++

Jenkins Master:

1.Jenkins Master perform basic installtion and handles task related to build and configuration.

2.schedule builds

3.they moniter slaves

4.records and presents the build resulits

Jenkins Slave:

\_\_\_\_\_\_\_\_\_\_\_-

1. Slaves are basically setup to offload builds from the master and disturbute the work loads

2. They listen to the master request

3. Slaves can run on variety of OS.(Linux & windows)

4. They minely execute bild jobs which are dispached by jenkins master instance jobs

Jenkins:

CI

scm:

git, svn, cvs,TFS

SVN - Plugin needs to be in the jenkins

Master and Slave Configuration:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Jenkins Pipeline :

2 type of pipeline jobs

1. Scripted pipeline - node (old )

2. Declarative Pipeline -

Groovy script we are using declarative pipilene

2 ways

- pipeline script - Groovy script

- Snippest generator - jenkins jobs dash board

5 Stages:

++++++++++

1. Continous Downloading(github)

2. Continuous Build

3. Continuous Deployment- automatic/manual deployment

4. Continuous Test - test

5. Continuous Delivery - ( final production delivery)

Groovy Scripting:

+++++++++++++

1. Pipeline{

agent none

stages{

stage('Checkout from git'){

agent any

stpes{

sh 'https://github.com/SobhaReddy/myE2e05Aug.git'

}

}