**Devops ci/cd with Jenkins maven git and pipeline part-1**

* **Meaning of Jenkins:**
* Jenkins is an open-source automation server that is widely used to facilitate continuous integration and continuous delivery (CI/CD) in software development
* Jenkins developed by Java language.
* It is the heart of DevOps.
* It is running on port number 8080.
* Jenkins is a plugins-based tool.
* It can be installed on various platforms including Windows, macOS, and Linux.
* It is a CI/CD tool, and it is one of the tools in DevOps.

**Jenkins architecture:**

**Install java prerequisites for running Jenkins: (In windows)**

* Install Java:
* Go to browser
* In search bar enter oracle java
* Click on download java
* After set environment variables
* Go to environment variables
* Clcik on environment variables
* Double click on path
* Click on new
* Path paste here
* Click on ok.
* **Install jenkins: (In windows)**
* Go to browser
* Enter download and install jenkins setup
* Login into jenkins dashboard
* Click on select plugins to install
* Click on None
* Install
* Create first admin user(enter user name,password,confirm password, and full name)
* Click on save and finish
* Click on start using jenkins.
* **Meaning of git:**
* Git is a distributed version control system that enables multiple developers to collaborate on software projects efficiently.
* It is free and open source.
* It was created by Linux Torvalds in 2005.
* Git helps you keep track of code changes.
* It is version control system means it is also known as source control is the practice of tracking and managing changes to software code.

**Git states-three**

* Git hub account creation(completed)
* Git install(completed)
* After completion of installation
* We can set username and password.
* Git config --global user.name “Bhavya”
* Git config --global user.mail [bhavyasri@puropalecreations.com](mailto:bhavyasri@puropalecreations.com)
* This tells git who is making changes.
* **How to create repository and how to push file from local machine to remote**
* Login into github account
* Click on new
* Enter repository name
* Click on add README file
* Put it in public or private
* Click on new
* Repository will be created.
* **Push files from local to remote location**:
* Git init
* Git add --all
* Git commit -m “message”
* Git branch -M main
* Git remote add origin <repository URL>
* Git push -u origin main.
* **Introduction of maven:**
* It is an open-source build automation tool.
* Project management for java applications
* Primarily used for java applications.
* Automates the source code compilation, dependency management, assembling of binary codes into packages and executes the test scripts.
* It follows a conversion over configuration approach.
* It provides default project structures and lifecycle phases and reduces manual configurations.
* Firstly developers, develops the code, then pushes to the GitHub (repo), then devops engineer will turn the code into packages by using build tools, deploys into the server, the end product will be accessed by clients/end users.
* Maven developed by Apache tomcat.
* Packages of maven:
* War: web applications
* Jar: Java applications
* Ear: enterprise applications.
* Maven installation
* Prerequisites
* Java
* **Maven core concepts:**

1. **Pom.xml file:**

* Pom.xml stands for project object model and .xml means Xtensible markup language
* Must be present in the projects root directory
* Root element <project>
* Contains everything needed to build a project using maven.

1. **Maven coordinates:**

* The first few elements of pom.xml file makes maven coordinates.
* The first few elements groupid, artifactid, version from the coordinates.
* It marks a specific place in a repository and hence called coordinates.

1. **Maven goals:**

* A goal is an action that maven performs in a phase.
* A plugin is a collection of goals

1. **Maven plugins:**

Need to information more about single plugin mvn help :describe -Dplugin=compiler

1. **Maven lifecycle:**

Mvn clean

Mvn validate

Mvn compile

Mvn test

Mvn package

Mvn verify

Mvn install

Mvn deploy

1. **maven repositories:**

There are three repositories:

Remote

Central

Local

1. **Dependency management:**

This section contains main libraries the project depends on

* **Convention over configuration:**
* Every artifact of a project has its own place.
* Source files
* Test files
* Packaged jar/war files
* Pom.xml
* Installed files
* Repositories.