Jenkins CI CD

1) Build & Deployment process

2) Challenges with Manual build & deployment

3) Jenkins Introduction

4) Jenkins Setup

5) Jenkins CI CD Pipeline

6) Git + Maven + Docker + Jenkins

What is Build & Deployment

1) Take latest source code from git hub repo

2) Compile Source Code

3) Execute Unit Test cases

4) Package our application (jar/war)

5) Build Docker image

6) Create Docker container

Challenges in Manual build & deployment

1) Every day we need to deploy latest code

2) Deploy code in multiple environments

   Ex: DEV, SIT, UAT and PROD

3) Takes lot of time

4) Repeated Work

5) Error Prone

What is Jenkins ?

1) Jenkins is free & open source software

2) Jenkins developed using Java language

3) Jenkins is used to automate Build & Deployment process

4) Using Jenkins we can implement CI CD

=> CI CD means continuous integration and continous deployment.

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

Jenkins Setup
===============

https://github.com/ashokitschool/DevOps-Documents/blob/main/04-Jenkins-Docker-Project.md

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

Jenkins Pipeline
======================

=> Jenkins pipeline contains set of steps to automate project build and deployment process.

=> We can create jenkins pipeline in 2 ways

1) Declarative pipeline

2) Scripted Pipeline (groovy)

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

Jenkins Declarative Pipeline Syntax
=====================================

```
pipeline {

        agent any

        stages {
                stage('Hello'){
                        steps{
                                // logic
                        }
                }
        }
}
```

==

Git + Maven + Docker + Jenkins Integration
==

Git Hub Repo :: https://github.com/ashokitschool/maven-web-app

1) Configure maven as global tool in jenkins

   Global Tool Name : Maven-3.9.9

2) install docker in jenkins vm

3) Create CI CD pipeline in jenkins

   Stage-1 : Git Clone

   Stage-2 : Maven Build

   Stage-3 : Build Docker Image

   Stage-4 : Deployment

4) Execute CI CD pipeline & observe console output

5) Enable host port in ec2 vm security group inbound rules

6) Access our application in browser

URL : http://public-ip:host-port/maven-web-app/

================ Final CI CD Pipleline ===========================

```
pipeline {
    agent any

    tools{
        maven "Maven-3.9.9"
    }

    stages {
        stage('Git Clone') {
            steps {
                git 'https://github.com/ashokitschool/maven-web-app.git'
            }
        }

        stage('Maven Build'){
            steps{
                sh 'mvn clean package'
            }
        }

        stage('Build Docker Image'){
            steps{
                sh 'docker build -t ashokitapp .'
            }
        }

        stage('Deployment'){
            steps{
                sh 'docker run -d -p 9090:8080 --name ashokitapp ashokitapp'
            }
        }
    }
}
```

====================
Real-Time workflow
====================

=> DevOps team will setup Jenkins server in linux vm

=> DevOps team will manage users in jenkins server
(role based access)

=> For Development team members, only jobs read & execute access will be available.

=> Development team will send request to DevOps team to create jenkins pipeline for the project.

=> Based on Dev Team request, DevOps team will create CI CD pipeline for the project.

=> Dev Team members can run CI CD pipeline for project build & deployment process.

Note: If CI CD job execution got failed then we need to check job execution logs (console output).