```
############################
Project Setup using below tools
############################

1) Maven
2) Git Hub
3) Jenkins
4) Docker
5) Kubernetes


1) Installation of Jenkins

2) Installation of K8S cluster

3) Setup Jenkins Server To deploy applications into K8S cluster

4) Build & Deploy Docker applications from Jenkins in K8S cluster using Pipeline script.


-> Developer will create a project and push code into git repo


-> Jenkins Server workflow
                -> clone git repo
                -> execute mvn packages
                -> build docker image
                -> push docker image
                -> deploy docker image in k8s cluster

++++++++++++
Work flow
++++++++++++

######## Step-1 ########

1) Create Ubuntu VM using AWS EC2

2) Install Java & Jenkins using below commands

$ sudo apt-get update

$ sudo apt-get install openjdk-8-jdk

$ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

$ sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ >
/etc/apt/sources.list.d/jenkins.list'

$ sudo apt-get update

$ sudo apt-get install jenkins

$ sudo systemctl status jenkins

# Copy jenkins admin pwd
$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword

# Open jenkins server in browser using VM public ip

URL : http://public-ip:8080/

-> Finish Jenkins setup
```

```
####### Step-2 : Setup Docker in Jenkins ######

1) install docker

curl -fsSL get.docker.com | /bin/bash

2) Add Jenkins user to docker group

sudo usermod -aG docker jenkins

3. Restart Jenkins

sudo systemctl restart jenkins


###### Step-3 :: Setup K8S cluster ######

1) Create 1 Master node & 2 worker nodes

Note: We have already created k8s kubeadm cluster, we can use that


######## Step-4 : Create Jenkins Pipeline ######

->  Create pipeline script

#########################
stage-1 : git clone
#########################

-> generate pipeline syntax for git clone with credentials

git credentialsId: 'c87aff7e-f5f1-4756-978f-3379694978e6', url:
'https://github.com/ashokitschool/maven-web-app.git'

#########################
stage-2 : mvn clean build
#########################

-> install maven using global tool configuration

def mavenHome = tool name: "Maven-3.8.6", type: "maven"

def mavenCMD = "${mavenHome}/bin/mvn"

sh "${mavenCMD} clean package"

#################################################
stage-3 : build and push docker image
#################################################

stage ('Build Docker Image'){
       sh "docker build -t ashokit/mavenwebapp ."
}

#################################################
Stage-4 : Push docker image into docker hub
#################################################

-> push docker image into docker hub using secret text
-> Use pipeline syntax to generate secret for docker hub account
```

```
stage ('Docker Push'){
       withCredentials([usernameColonPassword(credentialsId: 'DOCKER-HUB-CREDENTIALS', variable:
'DOCKER')]) {

       sh "docker login -u ashokit -p ${DOCKER_CREDENTIALS}"
    }
       sh "docker push ashokit/mavenwebapp"
}

##########################

-> Install Kubernetes Continuous Deployment plugin in Jenkins

-> Copy kube config file from k8s master node and add credential in Jenkins Global Credentials

ls .kube

cat .kube/config

##########################
Step-5 : Deploy in k8s
##########################

######################
Install kubectl in Jenkins Server
#################################

$ curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

$ chmod +x ./kubectl

$ sudo mv ./kubectl /usr/local/bin/kubectl

$ kubectl version

$ mkdir ~/.kube

Note: Take kube config file from k8s master node using below command and copy that
$ cat ~/.kube/config

# Paste Master Node Kube config file data here
$ vi ~/.kube/config

$ kubectl cluster-info

$ kubectl get nodes

######################## Final Script ##############################
node {

    stage ('Git Clone'){
        git credentialsId: 'c87aff7e-f5f1-4756-978f-3379694978e6', url:
'https://github.com/ashokitschool/maven-web-app.git'
    }

    stage ('Maven Clean Build'){
        def mavenHome = tool name: "Maven-3.8.6", type: "maven"
        def mavenCMD = "${mavenHome}/bin/mvn"
        sh "${mavenCMD} clean package"
    }

    stage ('Build Docker Image'){
       sh "docker build -t ashokit/mavenwebapp ."
    }
```

```
     stage ('Docker Push'){
        withCredentials([string(credentialsId: '278c300d-d845-4989-9169-0a00ea37a64c', variable:
 'DOCKER_CREDENTIALS')]) {
        sh "docker login -u ashokit -p ${DOCKER_CREDENTIALS}"
     }

         sh "docker push ashokit/mavenwebapp"
     }

    stage('Deploy App in K8S Cluster'){
        sh 'kubectl apply -f maven-web-app-deploy.yml'
     }

 }
```