**Tool which are going to use in this Project:**

* **Jenkins**
* **Docker**
* **Kubernetes**
* **AWS**

**Pre-Requesting:**

**Setup Two Server:**

1. **One for Jenkins**
2. **One for Kubernetes Client**

**Tools Installation and Pre-questing Setup:**

1. **Install Jenkins Tool on Jenkins Server**

**(or)**

**We can install the Jenkins while launching the instance by using user data.**

**#!/bin/bash**

**sudo yum update -y**

**sudo wget -O /etc/yum.repos.d/jenkins.repo \**

**https://pkg.jenkins.io/redhat-stable/jenkins.repo**

**sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key**

**sudo dnf install java-17-amazon-corretto -y**

**sudo yum install jenkins -y**

**sudo systemctl enable jenkins**

**sudo systemctl start jenkins**

1. **Install Kubernetes Cluster Setup by using KOPS Method on Kubernetes Client Machine.**
2. **Install Maven, Git, Docker on the Jenkins Server.**

**$ sudo yum install docker -y**

**$ systemctl start docker**

**$ yum install maven -y**

**$ yum install git -y**

1. **Adding Jenkins user on the Docker group**

**$ sudo usermod -a -G docker Jenkins**

**$ sudo systemctl restart Jenkins**

**Setup:**

* **Go to Jenkins dashboard**
* **New Item**
* **Create job Name**
* **Click Pipeline -> ok**

**Start to write the Jenkins Declarative pipeline:**

**Stage-1(git checkout):**

**Click Sample Hello World and edit for project requirement**

**For git logic we use pipeline syntax**

* **Sample Step -> git: Git**
* **Copy the git hub Repository URL**
* **Generate Pipeline Script-> copy the logic script and past in Jenkins pipeline**

**pipeline {**

**agent any**

**stages {**

**stage('git checkout') {**

**steps {**

**git 'https://github.com/arunkeerthi/newmavenproject.git'**

**}**

**}**

**}**

**}**

**Stage-2(git checkout+Maven):**

**pipeline {**

**agent any**

**stages {**

**stage('git checkout') {**

**steps {**

**git 'https://github.com/arunkeerthi/newmavenproject.git'**

**}**

**}**

**stage('mvn build') {**

**steps {**

**sh 'mvn clean package'**

**}**

**}**

**}**

**}**

**Stage-3(git checkout+Maven+docker image):**

**pipeline {**

**agent any**

**stages {**

**stage('git checkout') {**

**steps {**

**git 'https://github.com/arunkeerthi/newmavenproject.git'**

**}**

**}**

**stage('mvn build') {**

**steps {**

**sh 'mvn clean package'**

**}**

**}**

**stage('docker build') {**

**steps {**

**sh 'docker build -t arunkeerthi3101/demo1 .'**

**}**

**}**

**}**

**}**

**Stage-4(git checkout+Maven+docker image + docker login and docker push):**

**🡺Go to Pipeline Syntax -> withCredintials: Bind credentials to variables**

**🡺Add**

**🡺Secret Text**

**🡺Variable -> docker\_hub (you can give name)**

**🡺Click Add -> select “Jenkins”**

**🡺Click kind -> select “secret Text”**

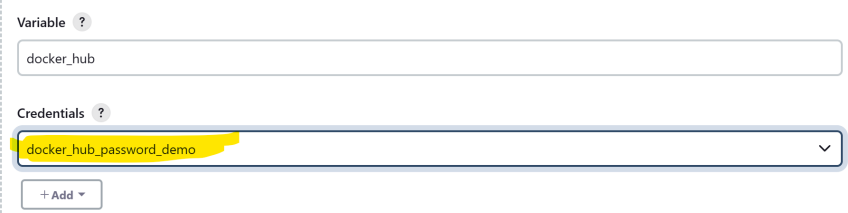
**🡺Secret -> (mention the docker hub password)**

**🡺ID->** **docker\_hub\_password (you can give any name)**

**🡺Description -> docker\_hub\_password\_demo (you can give any name for your reference)**

**🡺Click Add**

Note: Make sure (credentials which created update or not)

****

**🡺Click Generate Pipeline:**

**Copy the logic and past it in the Jenkins pipeline, then delete “some block “give comments as mentioned below**

**pipeline {**

**agent any**

**stages {**

**stage('git checkout') {**

**steps {**

**git 'https://github.com/arunkeerthi/newmavenproject.git'**

**}**

**}**

**stage('mvn build') {**

**steps {**

**sh 'mvn clean package'**

**}**

**}**

**stage('docker build') {**

**steps {**

**sh 'docker build -t arunkeerthi3101/demo1 .'**

**}**

**}**

**stage('docker login and push') {**

**steps {**

**withCredentials([string(credentialsId: 'docker\_hub\_password', variable: 'docker\_hub')]) {**

**sh 'docker login -u arunkeerthi3101 -p ${docker\_hub}'**

**sh 'docker push arunkeerthi3101/demo1'**

**}**

**}**

**}**

**}**

**}**

**Stage-5(git checkout+Maven+docker image + docker login and docker push + Login into k8 and copy the file):**

**# Jenkins server needs to communicate with Kubernetes Cluster**

**# we need to copy and past the Kubernetes spec file from Jenkins server to K8 Cluster Machine**

**# Need to update the. kubeconfig from root user to /home/ubuntu user**

**# Needs to install the SSH Plugin in Jenkins server for Communication between two servers**

* **Manage Jenkins**
* **Plugin**
* **Available Plugin**
* **SSH Agent -> install**

**Go to Pipeline syntax -> ssh agent:SSH Agent ->Add -> Jenkins -> sshusername with private key->ID(K8\_client)we can give any name -> Description(K8\_client\_demo1)we can give any name-> username(root)->Enter Directly(copy and paste the key)-> generate pipeline**

**Copy and past the (logic on the Jenkins pipline)**

**Stage('k8 login') {**

**steps {**

**sshagent(['K8\_client']) {**

**sh 'scp -o StrictHostKeyChecking=no ./maven-web-app-deploy.yml ubuntu@3.17.180.24:/home/ubuntu'**

**}**

**}**

**}**

**Note: Ensure that Kube config should updated in the /home/ubuntu username. For config follow below command**

**$ cp -r .kube /home/ubuntu**

**Stage-6(git checkout+Maven+docker image + docker login and docker push + Login into k8 and copy the file+ Deployment into k8):**

**stage('k8 deployment command') {**

**steps {**

**sshagent(['K8\_client']) {**

**sh 'ssh ubuntu@3.17.180.24 "sudo kubectl apply -f maven-web-app-deploy.yml"'**

**sh 'ssh ubuntu@3.17.180.24 "sudo kubectl rollout restart deployment/mavenwebappdeployment"'**

**}**

**}**

**}**

**Note : Make sure note the deployment name for rollout command**

**Final Pipeline**

**pipeline {**

**agent any**

**stages {**

**stage('git checkout') {**

**steps {**

**git 'https://github.com/arunkeerthi/newmavenproject.git'**

**}**

**}**

**stage('mvn build') {**

**steps {**

**sh 'mvn clean package'**

**}**

**}**

**stage('docker build') {**

**steps {**

**sh 'docker build -t arunkeerthi3101/demo1 .'**

**}**

**}**

**stage('docker login and push') {**

**steps {**

**withCredentials([string(credentialsId: 'docker\_hub\_password', variable: 'docker\_hub')]) {**

**sh 'docker login -u arunkeerthi3101 -p ${docker\_hub}'**

**sh 'docker push arunkeerthi3101/demo1'**

**}**

**}**

**}**

**stage('k8 login') {**

**steps {**

**sshagent(['K8\_client']) {**

**sh 'scp -o StrictHostKeyChecking=no ./maven-web-app-deploy.yml ubuntu@3.17.180.24:/home/ubuntu'**

**}**

**}**

**}**

**stage('k8 deployment command') {**

**steps {**

**sshagent(['K8\_client']) {**

**sh 'ssh ubuntu@3.17.180.24 "sudo kubectl apply -f maven-web-app-deploy.yml"'**

**sh 'ssh ubuntu@3.17.180.24 "sudo kubectl rollout restart deployment/mavenwebappdeployment"'**

**}**

**}**

**}**

**}**

**}**

**How to trigger one job to another job**

**🡺Go to pipline syntax 🡺 build:Build a job🡺Project to Build(Cd job name)**