pipeline {

environment {

registry = "sreeharshav/k8sb05"

registryCredential = 'DockerHubAccess'

dockerImage = ''

}

agent any

stages {

stage('Code Checkout') {

steps {

checkout([$class: 'GitSCM', branches: [[name: '\*/master']], extensions: [], userRemoteConfigs: [[credentialsId: 'GitHubAccess', url: 'git@github.com:SreeVeerDevOps/k8sb05-jenkins-integration.git']]])

sh "ls -al"

}

}

stage("Build Docker Image & Push") {

steps {

script { //add Scripted Pipeline sentences into a Declarative one

dockerImage = docker.build registry + ":$BUILD\_NUMBER"

docker.withRegistry( '', registryCredential ) {

dockerImage.push()

}

}

}

}

stage("Replace Image Name in Deployment.yaml") {

steps {

sh "yq e '.spec.template.spec.containers[0].image = \"sreeharshav/k8sb05:$BUILD\_NUMBER\"' -i deployment.yaml"

sh "cat deployment.yaml"

}

}

stage("Deploying Kubernetes") {

steps {

sh "kubectl apply -f deployment.yaml -f service.yaml"

}

}

}

}

pipeline {

environment {

dockerImage = ''

}

agent any

stages {

stage('Code Checkout') {

steps {

checkout([$class: 'GitSCM', branches: [[name: '\*/master']], extensions: [], userRemoteConfigs: [[credentialsId: 'GitHubAccess', url: 'git@github.com:SreeVeerDevOps/k8sb06-jenkins-ado-integration.git']]])

sh "ls -al"

}

}

stage('Kubectl Apply Dry Run') {

steps {

sh "kubectl apply -n ${ENVR} -f voting-app-aws.yml -o yaml --dry-run"

}

}

stage('Kubectl Apply') {

steps {

sh "kubectl apply -n ${ENVR} -f voting-app-aws.yml"

}

}

}

}

https://sanderh.dev/parsing-YAML-files-using-yq/

#Installing yq for YAML Parsing:

sudo add-apt-repository ppa:rmescandon/yq

sudo apt-get install yq -y

#Will show the file once the value repalced but wont actually replace the cvalue.

yq e '.spec.template.spec.containers[0].image = "sreeharshav/k8sb05:$BUILD\_NUMBER"' nginx.yaml

#Will replace the value in the file.

yq e '.spec.template.spec.containers[0].image = "sreeharshav/k8sb05:$BUILD\_NUMBER"' -i deployment.yaml