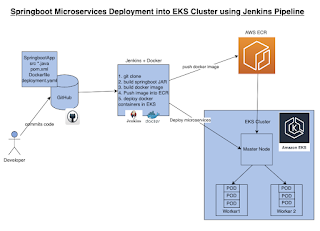
**[Deploy Springboot Microservices App into Amazon EKS Cluster using Jenkins Pipeline and Kubectl CLI Plug-in | Containerize Springboot App and Deploy into EKS Cluster using Jenkins Pipeline](https://www.coachdevops.com/2022/01/deploy-springboot-microservices-app_11.html)**

We will learn how to automate springboot microservices builds using Jenkins pipeline and Deploy into AWS EKS Cluster with help of Kubernetes CLI plug-in.

We will use Springboot Microservices based Java application. I have already created a repo with source code + Dockerfile. The repo also have Jenkinsfile for automating the following:

- Automating builds using Jenkins  
- Automating Docker image creation  
- Automating Docker image upload into AWS ECR  
- Automating Docker Containers Deployments to Kubernetes Cluster

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEg-CDo3r0LWddE5FLOq1w9K-N7MFfosRsf0jFVmr8oEdWRn4_NTHhgQ2eciSen6rghNi_rPWZ2N4BRTWxM9y10mhi5laXDIC5LrNY5sLBgqlxe08qzMsrMI_y3Np7VUschr79SeMuaBLiBZtiYRjMLJnw6LBbLdJV-pnb0_IGPd9yx0htmp2c7dgrqb/s899/Screen%20Shot%202022-09-04%20at%201.33.59%20PM.png)

**Same Code for this video is here:**

Make sure you fork my repo <https://github.com/akannan1087/springboot-app>

**Pre-requistes:**  
1. Amazon EKS Cluster is setup and running. Click [here](https://www.coachdevops.com/2022/02/create-amazon-eks-cluster-by-eksctl-how.html) to learn how to create Amazon EKS cluster.

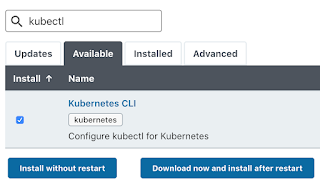
2. [Create ECR repo in AWS](https://www.cidevops.com/2020/05/how-to-setup-elastic-container-registry.html)

3. [Jenkins Master is up and running](https://www.coachdevops.com/2020/04/install-jenkins-ubuntu-1804-setup.html)

4. [Docker installed on Jenkins instance](https://www.coachdevops.com/2020/05/docker-jenkins-integration-building.html)

5. Docker, Docker pipeline and Kubernetes CLI plug-ins are installed in Jenkins

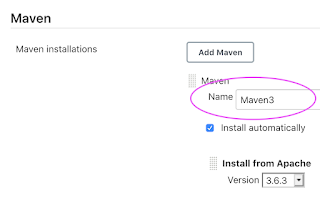
[](https://1.bp.blogspot.com/-ub6WkGJ-Op0/Xt_vpFJ6seI/AAAAAAAACik/2-bcHRPt-GQuKMFTPXMl17cHoPFI52AYQCK4BGAsYHg/s920/docker-plugins.png)

[](https://blogger.googleusercontent.com/img/a/AVvXsEiNDaWR1hHRZOt6G3OeUZS2EQOoNdthqj18R-kcAudsSwO7--SJ8d22J7ujPg-R5Xq8IIo9bP9XAXoWlp1ze8WB20LWl5mu0-WuPpWIKj6eEYwumv1POeS1Ax1vGVnhd0yMVSHW0DPhULNWI-DfxAk3hOYtdbTnYDozvEg8auvs4XjWGuPj5gWDZSmW=s320)

6. [Install kubectl](https://www.coachdevops.com/2020/10/install-kubectl-on-ubuntu-instance-how.html) on your instance

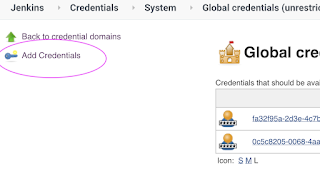
**Step # 1 - Create Maven3 variable under Global tool configuration in Jenkins**

Make sure you create Maven3 variable under Global tool configuration.

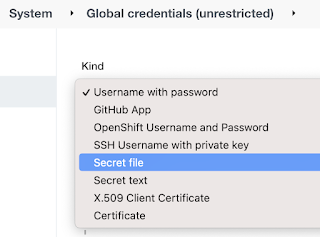
[](https://1.bp.blogspot.com/-qqqOVip7K6I/YEAxDxYsALI/AAAAAAAADUo/gqlv4lsl4x0u1lT8wf87S637h-NJHhrdwCLcBGAsYHQ/s968/maven3.png)

**Step #2 - Create Credentials for connecting to Kubernetes Cluster using kubeconfig**

Click on Add Credentials, use Kubernetes configuration from drop down.

[](https://1.bp.blogspot.com/-Mvy6LFnLTsY/Xq-NSobh4VI/AAAAAAAACFc/Pkx9j31nxKobJySQegbx2uDD36JSNrHuACLcBGAsYHQ/s1600/add.png)

use secret file from drop down.

[](https://blogger.googleusercontent.com/img/a/AVvXsEj87Lftt4azdQg8eugpMj1YAmu9XoNC-oJsqMr5AHJMOL_dFGqIw5fzADKTocJ-E-u9KMnaDW8lrA431jm_mq0Z9R9zBRxiwIFAIuSyDGkVclUr-UnL1Bpt_O0fymsuV9GwbYyJdGLG1BPRpK8QmqZsyT6x_2CpIioRt5I3Z_-rv5jPInF49tXsjghaIw=s834)

execute the below command to login as jenkins user.

sudo su - jenkins

you should see the nodes running in EKS cluster.

kubectl get nodes

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEiWKwQKkodHDUTMwiiduNmlDg0CbpkRvPD7xQLIizKvnoMroaeIq1Di1YsJ17cpE5aSlpMtnMWOzAZNiowSIhygwdc-Z8USCwDPNO57q2t-8WfuAfjqOAqXunJ9OgKwd1ts7J4okIWg5BQ0VLcR6sEwV5G1r3Bl0tLSAwUffFq9xU-FwqM4g1EpQvSj/s1600/Screen%20Shot%202022-05-13%20at%207.26.51%20PM.png)

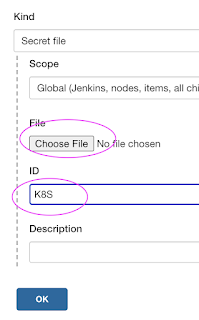
Execute the below command to get kubeconfig info, copy the entire content of the file:

cat /var/lib/jenkins/.kube/config

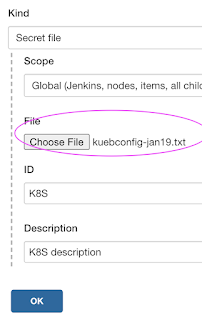
[](https://blogger.googleusercontent.com/img/a/AVvXsEh07GG01S0ezTnehqzoCjlHugN94T9-xwwv2ErJPG0I268sDgSANIMYcfFxV2n84TcSNos7FVprINGC_NlXiCBS5_Pgfybv81ccRlwPaUDO9AJVeU85ibtaD6VMlQ3OhptxEgqpxTmT9Hv3lcPxCAxkw3Uxc3kjLJnaj5QwnQLLg16aDLaDILUE5zZ4=s2376)

Open your text editor or notepad, copy and paste the entire content and save in a file.

We will upload this file.

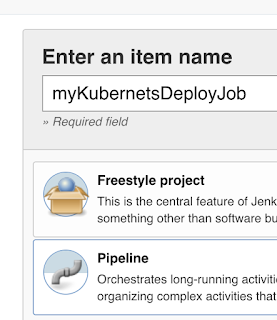
[](https://blogger.googleusercontent.com/img/a/AVvXsEgNnKliu1KL-cn_Ze5GgmrzCX6TUfs_TUC5xKu0P3q-QhfaN97pV_DRmZ-4Sh-zrbHMEl--CwHHC1MzJfyk9fh0ezzXOcExilGR6rVPsNISfEfXNadRplfeY9z3sAJ8ErXiBr5mx7I1OXO2CAH0b70tP1873KzRjLI5jzq2NW1T8WJPltq06bLGi_tACQ=s896)

Enter ID as K8S and choose File and upload the file and save.

[](https://blogger.googleusercontent.com/img/a/AVvXsEitDtvjqpqYnopwU3Oi4C46zEST16kIPVW8MtIeRStLWPe3Lfhx9pNTMnAXRWV3mdBF5ThlakzAGEGmmgbFzhbvIQJSxO_MTxbljAcAaOBWwa4nd1IdYvTtNEt4pTwiBqkNZzs0FLS1gq7on8TAS2vaDbVJSDuLYKXmkFtAE21BSiFMAL9LGo0LApmenw=s884)

Enter ID as K8S and choose enter directly and paste the above file content and save.

**Step # 3 - Create a pipeline in Jenkins**

Create a new pipeline job.  
[](https://1.bp.blogspot.com/-pk3H4-qXqUg/Xt_xYAAnbiI/AAAAAAAACjw/tIymD_gtPUsKFOn72WsjlUTHkGJHT7fXACK4BGAsYHg/s682/myk8sjob.png)

**Step # 4 - Copy the pipeline code from below**  
Make sure you change red highlighted values below as per your settings:  
Your docker user id should be updated.  
your registry credentials ID from Jenkins from step # 1 should be copied

pipeline {

   tools {

        maven 'Maven3'

    }

    agent any

    environment {

        registry = "account\_id.dkr.ecr.us-east-2.amazonaws.com/my-docker-repo"

    }

    stages {

        stage('Cloning Git') {

            steps {

                checkout([$class: 'GitSCM', branches: [[name: '\*/main']], doGenerateSubmoduleConfigurations: false, extensions: [], submoduleCfg: [], userRemoteConfigs: [[credentialsId: '', url: '<https://github.com/akannan1087/springboot-app>']]])

            }

        }

      stage ('Build') {

          steps {

            sh 'mvn clean install'

            }

      }

    // Building Docker images

    stage('Building image') {

      steps{

        script {

          dockerImage = docker.build registry

        }

      }

    }

    // Uploading Docker images into AWS ECR

    stage('Pushing to ECR') {

     steps{

         script {

                sh 'aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin account\_id.dkr.ecr.us-east-2.amazonaws.com'

                sh 'docker push account\_id.dkr.ecr.us-east-2.amazonaws.com/my-docker-repo:latest'

         }

        }

      }

       stage('K8S Deploy') {

        steps{

            script {

                withKubeConfig([credentialsId: 'K8S', serverUrl: '']) {

                sh ('kubectl apply -f  eks-deploy-k8s.yaml')

                }

            }

        }

       }

    }

}

**Step # 5 - Build the pipeline**

Once you create the pipeline and changes values per your configuration, click on Build now:

[](https://blogger.googleusercontent.com/img/a/AVvXsEi_H6PgEVPk7Cu8iikw6Ac8wtfABBU5AjyimjzRCSCkoF4ePKYSXCljhlL5OeBMC6WczqwXZogi-t_y5F_zMe7Mn0zKMFKLdQ-lNfaFTkvRYOtQeAKBTmp45S8h2omnPNdc-MUj_p2K1aNK2da4fLB-cPSe9o50qHYc50j4Y29AJV7c8r3NKzDRyHnL=s1628)

**Step # 6 - Verify deployments to K8S**

kubectl get pods

[](https://1.bp.blogspot.com/-xPmhS4ehKKM/X9lOutozp5I/AAAAAAAADLc/8EotGJ1lUOkLB0I0riVX9qVNhsUZhZUbgCLcBGAsYHQ/s583/podd2.png)

kubectl get deployments

[](https://1.bp.blogspot.com/-3rBuEPQN6JE/X9lOuq2G65I/AAAAAAAADLY/741eWCCWbzAvvb9d4WisbDYVi9tEPZqHwCLcBGAsYHQ/s535/d1.png)

kubectl get services

[](https://1.bp.blogspot.com/-Uwqj5y55nmU/X9lOSTh-xVI/AAAAAAAADLQ/lEJl0Yynvh0ZQSB71pBaiy87Qs74duhywCLcBGAsYHQ/s974/angular.png)

If you see any errors after deploying the pods, you can check the pod logs.

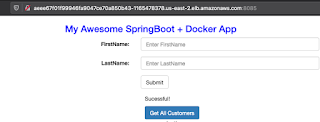
kubectl logs <pod\_name>

**Steps # 7 - Access SpringBoot App in K8S cluster**

Once build is successful, go to browser and enter master or worker node public ip address along with port number mentioned above

http://loadbalancer\_ip\_address

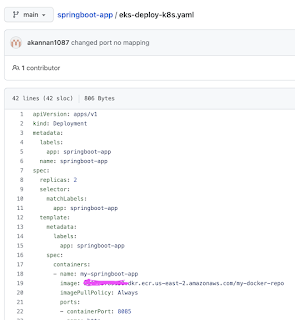
You should see page like below:

[](https://1.bp.blogspot.com/--2ejPZXHY2E/X9lNvGhHFkI/AAAAAAAADLI/Pk16EI2pq-YU5BXp8hl3TwLyKffUdyszwCLcBGAsYHQ/s822/springboot-app.png)

**Note:**

Make sure you fork my repo <https://github.com/akannan1087/springboot-app>

and make changes in eks-deploy-k8s.yaml to pull Docker image from your AWS ECR repo.

[](https://blogger.googleusercontent.com/img/a/AVvXsEiJeBcgLoKDVbh3koKvGW_ZzJT950fZ1p5ZROELylsvrwKqrO4W1AAUW1eDtdTjMnYjXJiAK1cfOqGYTywag-bE7RizNXlcFZe9RAGWlNrNirVwzfDpnoZ5CbXZZIoM7Pa_R2PM_FCiXWyyPPf8ZcYrclbCAK5fUPcUhR95NmFdTc0MNuCmwtRkzqPT=s1362)