**Task Description:**

**Problem:**

To create and configure a simple CI/CD pipeline using Jenkins or any other open-source automation tool. You can use: <https://github.com/postmanlabs/httpbin>

**The pipeline should be able to:**

* Get triggered when any new commit/merge is done to the Version Control System.
* Source the code from the VCS.
* Create the docker image and should update it on the machine where the deployment is live.

**Solution:**

**Steps Performed:**

I have deployed Jenkins and Kubernetes over AWS Cloud.

**For creating Pipeline:**

* Forked <https://github.com/postmanlabs/httpbin> Repo.
* As our pipeline will have to get triggered when any new commit/merge is done to the Version Control System for this we have to add WebHook in Github repo.
* After adding webhooks we will be creating a Pipeline.
* We will be writing a Pipeline script.

GitHub Repo: <https://github.com/Uditanshu0110/httpbin.git>

Also, I have uploaded an Yaml file for launching the pods and for deployment.

**Pipeline Script is described below.**

node {

stage('Git Clone') {

git credentialsId: 'Github\_Cred', url: 'https://github.com/Uditanshu0110/httpbin.git'

}

stage('Docker Build') {

sh 'docker build . -t uditanshu/newimage'

}

stage('Docker Push') {

withCredentials([string(credentialsId: 'Docker\_Sec', variable: 'Docker\_Sec')]) {

sh 'docker login -u uditanshu -p ${Docker\_Sec}'

}

sh 'docker push uditanshu/newimage'

}

stage('Kubernetes Deploy') {

sh 'kubectl apply -f pod.yml'

}

}

In this script we will be Cloning the Github repo to our Jenkins Workspace, Building Docker Image, Pushing it to Docker Hub, and Finally deploying it to Kubernetes.

**stage('Git Clone') {**

**git credentialsId: 'Github\_Cred', url: 'https://github.com/Uditanshu0110/httpbin.git'**

**}**

This block will clone the Github repo to Jenkins master. Here I have provided the link to the repo. And also added the Github user name and Github Password in Jenkins.

**stage('Docker Build') {**

**sh 'docker build . -t uditanshu/newimage'**

**}**

This block will build the docker image with the given tag name.

**stage('Docker Push') {**

**withCredentials([string(credentialsId: 'Docker\_Sec', variable: 'Docker\_Sec')]) {**

**sh 'docker login -u uditanshu -p ${Docker\_Sec}'**

**}**

**sh 'docker push uditanshu/newimage'**

**}**

This block will push the docker image in the docker hub account. Here I have added the docker password as a secret text in the Jenkins and used that variable after that pushed the created image in the docker hub account.

**stage('Kubernetes Deploy') {**

**sh 'kubectl apply -f pod.yml'**

**}**

This block will deploy the image in the Kubernetes cluster.

Pod.yml file used for Deployment and service.

**Components used.**

**I have used**

* AWS Cloud- For Setting up the Jenkins and Kubernetes Over the cloud.
* Jenkins.
* Docker.
* Git/Github.
* Kubernetes- For deploying.

**Jenkins URL -** <http://13.235.104.127:8080/>

**Username: admin**

**Password: redhat**

**The URI of the application hosted:**

As, I have hosted application over Kubernetes, So, by using Kubernetes Slave node Ip and port number of service we can see the application hosted on the Kubernetes.

**Current URL of Hosted application:**

<http://13.233.103.47:31604/>

**Some Images of tasks:**



