Business Requirements

• The team of developers working on new features will merge their code to a GitHub repo.

• As soon as the code reaches GitHub, using a CI (Continuous Integration) pipeline, setup in Jenkins, automated builds will be triggered.

• The automated builds will frequently deploy new features to the production website.

•Every build will prepare a Dockerfile and push docker images to docker-hub.

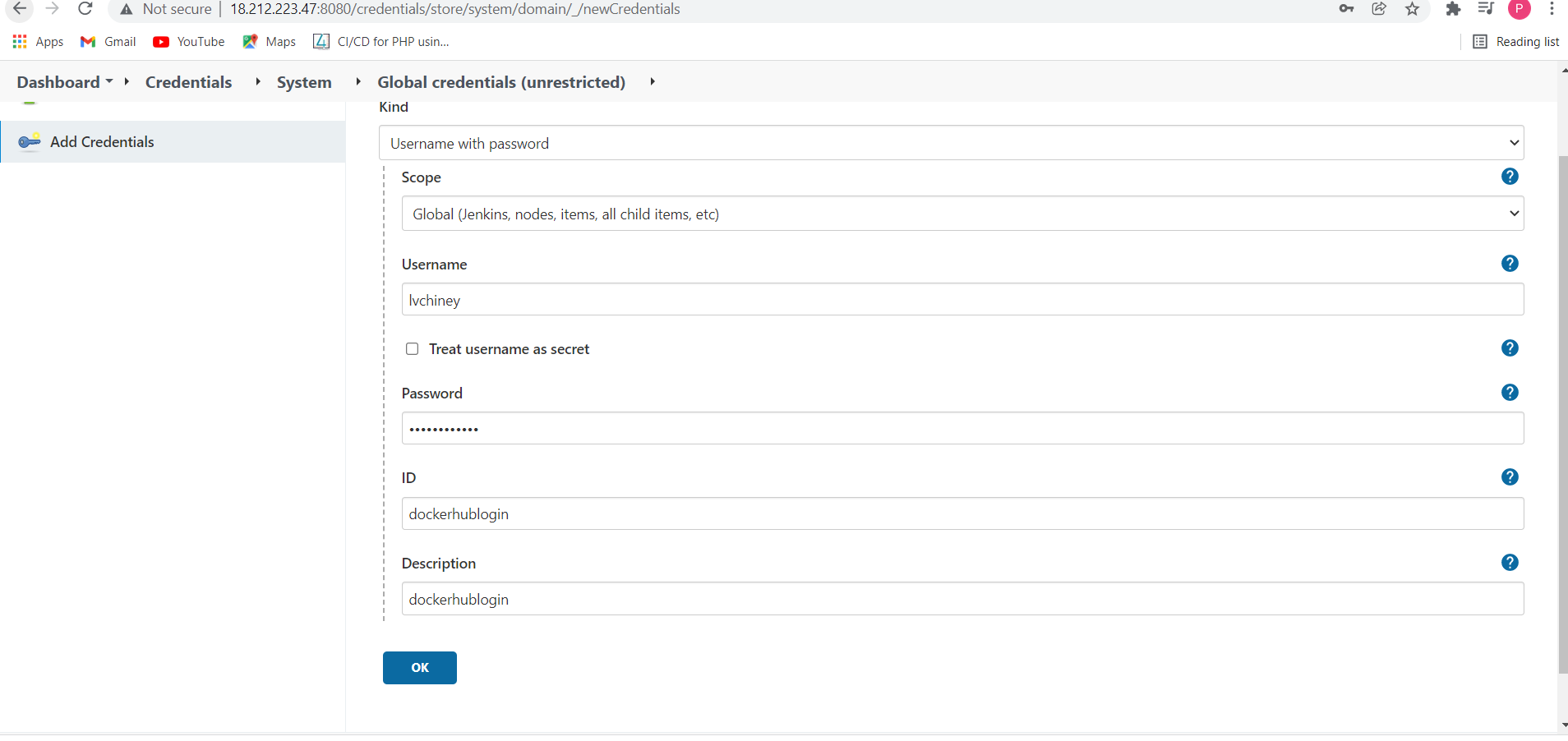
•Every docker image will be deployed (Continuous Deployment) to a kubernetes-cluster

GitHub: <https://github.com/lvchiney/cicd-pipeline-train-schedule-autodeploy.git>

Solution:

Install Kubernetes Continuous Plugin 1.0 version

Install Docker Pipeline Plugin



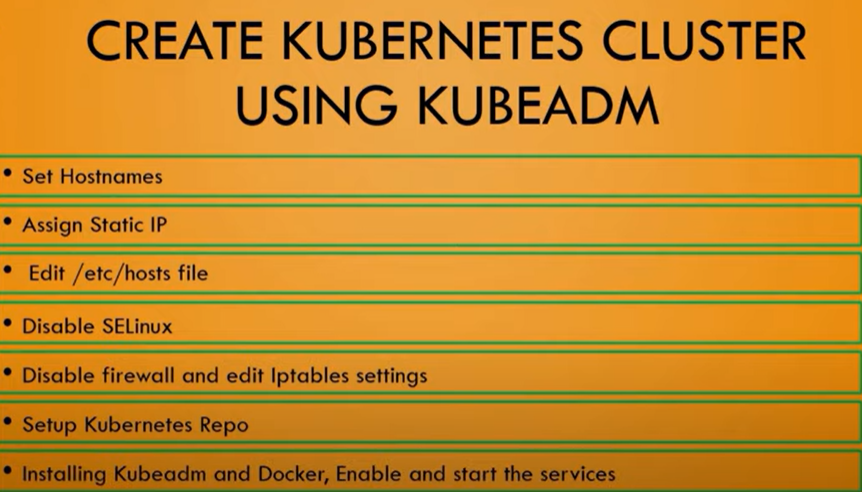
Set Up Kubernestes cluster (Master and Worker1)

Install docker in server

# This script is meant for quick & easy install via:

# $ curl -fsSL https://get.docker.com -o get-docker.sh

# $ sh get-docker.sh







docker image ls

[centos@ip-172-31-82-109 ~]$ sudo systemctl start docker

[centos@ip-172-31-82-109 ~]$ sudo systemctl enable docker

sudo docker image ls

sudo usermod -aG docker centos

sudo usermod -a -G docker jenkins

Pipeline:

pipeline {

environment {

dockerimagename = "lvchiney/project2"

dockerImage = ""

}

agent any

stages {

stage('Checkout Source') {

steps {

git 'https://github.com/lvchiney/cicd-pipeline-train-schedule-autodeploy.git'

}

}

stage('Build image') {

steps{

script {

dockerImage = docker.build dockerimagename

}

}

}

stage('Pushing Image') {

environment {

registryCredential = 'dockerhublogin'

}

steps{

script {

docker.withRegistry( 'https://registry.hub.docker.com', registryCredential ) {

dockerImage.push("latest")

}

}

}

}

stage('Deploying App to Kubernetes') {

steps {

script {

kubernetesDeploy(configs: "deploymentservice.yml", kubeconfigId: "kubernetes")

}

}

}

}

}

