Deploy to Kubernetes in Google Cloud: Challenge Lab  
  
  
Run all commands in Cloud Shell  
  
Task 1: Create a Docker image and store the Dockerfile

|  |
| --- |
| source <(gsutil cat gs://cloud-training/gsp318/marking/setup\_marking.sh)  gcloud source repos clone valkyrie-app  cd valkyrie-app  cat > Dockerfile <<EOF  FROM golang:1.10  WORKDIR /go/src/app  COPY source .  RUN go install -v  ENTRYPOINT ["app","-single=true","-port=8080"]  EOF  docker build -t valkyrie-app:v0.0.1 .  cd ~/marking  ./step1.sh |

Task 2: Test the created Docker image

|  |
| --- |
| cd valkyrie-app  docker run -p 8080:8080 valkyrie-app:v0.0.1 &  cd ~/marking  ./step2.sh |
|  |

Task 3: Push the Docker image in the Google Container Repository

|  |
| --- |
| cd valkyrie-app  docker tag valkyrie-app:v0.0.1 [gcr.io/$GOOGLE\_CLOUD\_PROJECT/valkyrie-app:v0.0.1](http://gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.1)  docker push [gcr.io/$GOOGLE\_CLOUD\_PROJECT/valkyrie-app:v0.0.1](http://gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.1) |
|  |

Task 4: Create and expose a deployment in Kubernetes

|  |
| --- |
| sed -i s#IMAGE\_HERE#[gcr.io/$GOOGLE\_CLOUD\_PROJECT/valkyrie-app:v0.0.1#g](http://gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.1#g) k8s/deployment.yaml  gcloud container clusters get-credentials valkyrie-dev --zone us-east1-d  kubectl create -f k8s/deployment.yaml  kubectl create -f k8s/service.yaml |

Task 5: Update the deployment with a new version of valkyrie-app

|  |
| --- |
| git merge origin/kurt-dev  kubectl edit deployment valkyrie-dev  ### change replicas from 1 to 3  docker build -t [gcr.io/$GOOGLE\_CLOUD\_PROJECT/valkyrie-app:v0.0.2](http://gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.2) .  docker push [gcr.io/$GOOGLE\_CLOUD\_PROJECT/valkyrie-app:v0.0.2](http://gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.2)  kubectl edit deployment valkyrie-dev  ### change 0.0.1 to 0.0.2 in two places  docker build -t [gcr.io/$GOOGLE\_CLOUD\_PROJECT/valkyrie-app:v0.0.2](http://gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.2) .  docker push [gcr.io/$GOOGLE\_CLOUD\_PROJECT/valkyrie-app:v0.0.2](http://gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.2) |

Task 6: Create a pipeline in Jenkins to deploy your app

|  |
| --- |
| docker ps  ### get container id  docker kill container\_id  export POD\_NAME=$(kubectl get pods --namespace default -l "[app.kubernetes.io/component=jenkins-master](http://app.kubernetes.io/component=jenkins-master)" -l "[app.kubernetes.io/instance=cd](http://app.kubernetes.io/instance=cd)" -o jsonpath="{.items[0].[metadata.name](http://metadata.name/)}")  kubectl port-forward $POD\_NAME 8080:8080 >> /dev/null &  printf $(kubectl get secret cd-jenkins -o jsonpath="{.data.jenkins-admin-password}" | base64 --decode);echo  # Open web-preview and login as admin with password from last command  # Click on "Manage Jenkins", then click on "Manage Credentials"  # Click on "Jenkins" in "Stores scopes to Jenkins", then click on "Global credentials(unrestricted)"  # Click on "Add Credentials" from the left menu.  # Select "Google Service Account from metadata" from kind and click on "OK".  # Click Jenkins (top left), then click new item and enter "valkyrie-app"  # Click on pipeline and then click “OK” and Select "pipeline script from SCM" in Definition and set SCM to Git  # Add the source code repo ( find it using command: gcloud source repos list)  # Set credentials to qwiklabs-...  # Click save.  # Change color  sed -i "s/green/orange/g" source/html.go  sed -i “s/YOUR\_PROJECT/$GOOGLE\_CLOUD\_PROJECT/g” Jenkinsfile (**If this command does not work, do the file changes manually by using nano or vi editor**)  # Update project in Jenkinsfile  git config --global user.email "[you@example.com](mailto:you@example.com)"  git config --global [user.name](http://user.name/) "student"  git add .  git commit -m "build pipeline init"  git push  # in jenkins click build now on the job  # initial build takes a while, just wait |