Kubernetes in Google Cloud: Challenge Lab

Task 1: Create a Docker image and store the Dockerfile

Open Cloud Shell and run `source <(gsutil cat gs://cloud-training/gsp318/marking/setup_marking.sh)`. This command will install marking scripts you can use to help check your progress.

Use Cloud Shell to clone the `valkyrie-app` source code repository (it is in your project).

The app source code is in `valkyrie-app/source`. Create `valkyrie-app/Dockerfile` and add the configuration below.

```
FROM golang:1.10
WORKDIR /go/src/app
COPY source .
RUN go install -v
ENTRYPOINT ["app","-single=true","-port=8080"]
```

Use `valkyrie-app/Dockerfile` to create a Docker image called **valkyrie-app** with the tag **v0.0.1**

Once you have created the Docker image, and before clicking **Check my progress**, run `step1.sh` to perform the local check of your work. After you get a successful response from the local marking you can check your progress.

TASK 1

source <(gsutil cat gs://cloud-training/gsp318/marking/setup_marking.sh)

```
student_00_7544f8e17b13@cloudshell:~ (qwiklabs-gcp-00-15c467e8f799)$ source <(gsutil cat gs://cloud-training/gsp318/marking/setup_marking.sh)
Copying gs://cloud-training/gsp318/marking/step1.sh...
Copying gs://cloud-training/gsp318/marking/step2.sh...
/ [2 files][  1.2 KiB/  1.2 KiB]
Operation completed over 2 objects/1.2 KiB.
```

gcloud source repos clone valkyrie-app

```
student_00_7544f8e17b13@cloudshell:~/marking (qwiklabs-gcp-00-15c467e8f799)$ gcloud source repos clone valkyrie-app
Cloning into '/home/student_00_7544f8e17b13/marking/valkyrie-app'...
remote: Total 41 (delta 7), reused 41 (delta 7)
Unpacking objects: 100% (41/41), done.
Project [qwiklabs-gcp-00-15c467e8f799] repository [valkyrie-app] was cloned to [/home/student_00_7544f8e17b13/marking/valkyrie-app].
```

cd valkyrie-app

cat > Dockerfile <<EOF

```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ cat > Dockerfile <<EOF
> FROM golang:1.10
> WORKDIR /go/src/app
> COPY source .
> RUN go install -v
> ENTRYPOINT ["app","-single=true","-port=8080"]
> 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 .

```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ docker build -t valkyrie-app:v0.0.1 .

Sending build context to Docker daemon  223.2kB
Step 1/5 : FROM golang:1.10
1.10: Pulling from library/golang
741437d97401: Pull complete
34d8874714d7: Pull complete
0a108aa26679: Pull complete
7f0334c36886: Pull complete
d35724ed4672: Pull complete
c0eaf021aeaf: Pull complete
d3d9c96611f1: Pull complete
Digest: sha256:6d5e79878a3e4f1b30b7aa4d24fb6ee6184e905a9b172fc72593935633be4c46
Status: Downloaded newer image for golang:1.10
 ---> 6fd1f7edb6ab
Step 2/5 : WORKDIR /go/src/app
 ---> Running in 51408678490d
Removing intermediate container 51408678490d
 ---> 1afdbf12fa01
Step 3/5 : COPY source .
 ---> e67141970580
Step 4/5 : RUN go install -v
 ---> Running in e5c34711f065
app/vendor/golang.org/x/net/context
app/vendor/golang.org/x/net/context/ctxhttp
app/vendor/cloud.google.com/go/compute/metadata
app
Removing intermediate container e5c34711f065
 ---> 5d9963e7d13f
Step 5/5 : ENTRYPOINT ["app","-single=true","-port=8080"]
 ---> Running in 2601a5e9ac4f
Removing intermediate container 2601a5e9ac4f
 ---> 9b410afa2fca
Successfully built 9b410afa2fca
Successfully tagged valkyrie-app:v0.0.1
```

step1.sh

```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ step1.sh
Image exists
Go ahead and check the activity tracking on the lab page
```

Task 2: Test the created Docker image

Launch a container using the image **valkyrie-app:v0.0.1**. You need to map the host's port 8080 to port 8080 on the container. Add `&` to the end of the command to cause the container to run in the background.

When your container is running you will see the page by **Web Preview**.

Once you have your container running, and before clicking **Check my progress**, run `step2.sh` to perform the local check of your work. After you get a successful response from the local marking you can check your progress.

docker run -p 8080:8080 valkyrie-app:v0.0.1 &

step2.sh

```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ docker run -p 8080:8080 valkyrie-app:v0.0.1 &
[1] 1382
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ 2020/05/03 07:37:39 Operating in single mode...

student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ step2.sh
Container running and visible on port 8080, good job!
Go ahead and check the activity tracking on the lab page
```

Task 3: Push the Docker image in the Container Repository

Push the Docker image **valkyrie-app:v0.0.1** into the Container Registry.

Make sure you re-tag the container to **gcr.io/YOUR_PROJECT/valkyrie-app:v0.0.1**.

TASK 3

- **docker tag valkyrie-app:v0.0.1 gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.1**
- **docker push gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.1**

```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ docker tag valkyrie-app:v0.0.1 gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.1
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ docker push gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.1
The push refers to repository [gcr.io/qwiklabs-gcp-00-15c467e8f799/valkyrie-app]
bf286d2fba44: Pushed
299f981193b0: Pushed
d8461f01c21a: Pushed
7b9a9415bf3a: Pushed
facf15440126: Pushed
77b4b6493272: Pushed
6257fa9f9597: Pushed
578414b395b9: Pushed
abc3250a6c7f: Pushed
13d5529fd232: Pushed

v0.0.1: digest: sha256:9f0d5327b169634c33fcfb6abb5418c5e4bc2baa445d149f21d56b0a3a0d7f49 size: 2423
```

Task 4: Create and expose a deployment in Kubernetes

Kurt created the `deployment.yaml` and `service.yaml` to deploy your new containe image to a Kubernetes cluster (called valkyrie-dev). The two files are in `valkyrie-app/k8s`.

Remember you need to get the Kubernetes credentials before you deploy the image onto the Kubernetes cluster.

Before you create the deployments make sure you check the `deployment.yaml` and `service.yaml` files. Kurt thinks they need some values set (he thinks he left some placeholder values).

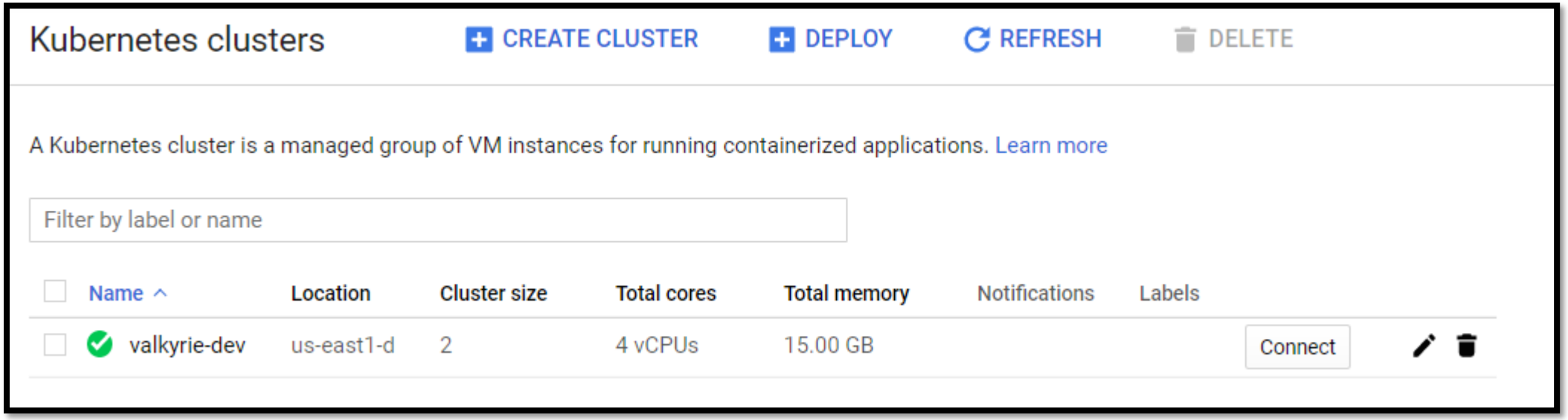
You can check the load balancer once it's available.

TASK 4

sed -i s#IMAGE_HERE#gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.1#g k8s/deployment.yaml

```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ sed -i s#IMAGE_HERE#gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.1#g k8s/deployment.yaml
```

- Kurbenetes Engine > Clusters
- Connect > copy
- gcloud container clusters get-credentials valkyrie-dev --zone us-east1-d --project qwiklabs-gcp-00-15c467e8f799




```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ gcloud container clusters get-credentials valkyrie-dev --zone us-east1-d --project qwiklabs-gcp-00-15c467e8f799
Fetching cluster endpoint and auth data.
kubeconfig entry generated for valkyrie-dev.
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$
```

kubectl create -f k8s/deployment.yaml

kubectl create -f k8s/service.yaml

```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ kubectl create -f k8s/deployment.yaml
deployment.extensions/valkyrie-dev created
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ kubectl create -f k8s/service.yaml
service/valkyrie-dev created
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$
```

TASK 5

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

Before deploying the new code, increase the replicas from 1 to 3 to ensure you don't cause an outage.

Kurt made changes to the source code (he put the changes in a branch called **kurt-dev**). You need to merge **kurt-dev** into **master** (you should use `git merge origin/kurt-dev`).

Build the new code as version v0.0.2 of valkyrie-app, push the updated image to the Container Repository, and then redeploy to the valkyrie-dev cluster. You will know you have the new v0.0.2 version because the titles for the cards will be green.

git merge origin/kurt-dev

```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ git merge origin/kurt-dev
Updating 655d315..ff1acc8
Fast-forward
 source/html.go | 4 ++--
 1 file changed, 2 insertions(+), 2 deletions(-)
```

kubectl scale deployment valkyrie-dev –replicas=3

```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ kubectl scale deployment valkyrie-dev --replicas=3
deployment.extensions/valkyrie-dev scaled
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$
```

kubectl get deployment

```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ kubectl get deployment
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
cd-jenkins     1/1     1            1           21m
valkyrie-dev   3/3     3            3           9m1s
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$
```

docker build -t gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.2 .

```
student_02_29ba9def961e@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-02-334faf351610)$ docker build -t gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.2 .
Sending build context to Docker daemon  241.2kB
Step 1/5 : FROM golang:1.10
 ---> 6fd1f7edb6ab
Step 2/5 : WORKDIR /go/src/app
 ---> Using cache
 ---> 79559f273bec
Step 3/5 : COPY source .
 ---> 53d7db5723be
Step 4/5 : RUN go install -v
 ---> Running in 60dd42febd1a
app/vendor/golang.org/x/net/context
app/vendor/golang.org/x/net/context/ctxhttp
app/vendor/cloud.google.com/go/compute/metadata
app
Removing intermediate container 60dd42febd1a
 ---> 00fd52ae491c
Step 5/5 : ENTRYPOINT ["app","-single=true","-port=8080"]
 ---> Running in ec329e2dddd7
Removing intermediate container ec329e2dddd7
 ---> 51f35d2d54de
Successfully built 51f35d2d54de
Successfully tagged gcr.io/qwiklabs-gcp-02-334faf351610/valkyrie-app:v0.0.2
```

docker push gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.2

```
student_02_29ba9def961e@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-02-334faf351610)$ docker push gcr.io/$GOOGLE_CLOUD_PROJECT/valkyrie-app:v0.0.2
The push refers to repository [gcr.io/qwiklabs-gcp-02-334faf351610/valkyrie-app]
620392056190: Pushed
a1844cde6ef4: Pushed
3d0dbfd2cd85: Layer already exists
7b9a9415bf3a: Layer already exists
facf15440126: Layer already exists
77b4b6493272: Layer already exists
6257fa9f9597: Layer already exists
578414b395b9: Layer already exists
abc3250a6c7f: Layer already exists
13d5529fd232: Layer already exists

v0.0.2: digest: sha256:28fbc08fab640d3f28d46f05401cd593e117c2ae8b955efbc3e196ab34e605de size: 2423
```

kubectl edit deployment valkyrie-dev

```
student_02_29ba9def961e@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-02-334faf351610)$ kubectl edit deployment valkyrie-dev
```

Edit:

v0.0.1 to v0.0.2

Vi

esc
:x (save and exit)
enter

TASK 6

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

This process of building the container and pushing to the container repository can be automated using Jenkins. There is a Jenkins deployment in your `valkyrie-dev` cluster - connect to Jenkins and configure a job to build when you push a change to the source code.

Remember with Jenkins:

- Get the password with `printf $(kubectl get secret cd-jenkins -o jsonpath="{.data.jenkins-admin-password}" | base64 --decode);echo`.
- Connect to the Jenkins console using the commands below (but make sure you don't have a running container `docker ps`; if you do, kill it):

```
export POD_NAME=$(kubectl get pods --namespace default -l
"app.kubernetes.io/component=jenkins-master" -l
"app.kubernetes.io/instance=cd" -o jsonpath="{.items[0].metadata.name}")
kubectl port-forward $POD_NAME 8080:8080 >> /dev/null &
```

- Setup your credentials to use **Google Service Account from metadata**.
- Create a pipeline job that points to your `*/master` branch on your source code.

Make two changes to your files before you commit and build:

- Edit `valkyrie-app/Jenkinsfile` and change YOUR_PROJECT to your actual project id.
- Edit `valkyrie-app/source/html.go` and change the two occurrences of green to orange.

Use `git` to:

- Add all the changes then commit those changes to the master branch.
- Push the changes back to the repository.

When you are ready, manually trigger a build (the initial build will take some time, so just monitor the process). The build will replace the running containers with containers with different tags; you will see orange colored headings.

docker ps

```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ docker ps
CONTAINER ID        IMAGE                  COMMAND                  CREATED             STATUS              PORTS                    NAMES
695d9ec193e2        valkyrie-app:v0.0.1    "app -single=true -p…"   40 minutes ago      Up 40 minutes       0.0.0.0:8080->8080/tcp   stupefied_chandrasekhar
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$
```

docker kill 695d9ec193e2

```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ docker kill 695d9ec193e2
695d9ec193e2
[1]-  Exit 137                docker run -p 8080:8080 valkyrie-app:v0.0.1
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$
```

export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/component=jenkins-master" -l "app.kubernetes.io/instance=cd" -o jsonpath="{.items[0].metadata.name}")

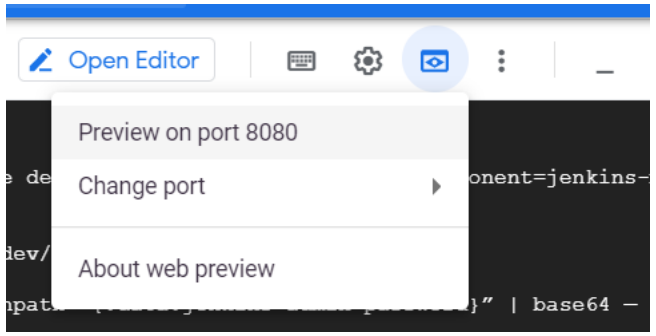
kubectl port-forward $POD_NAME 8080:8080 >> /dev/null &

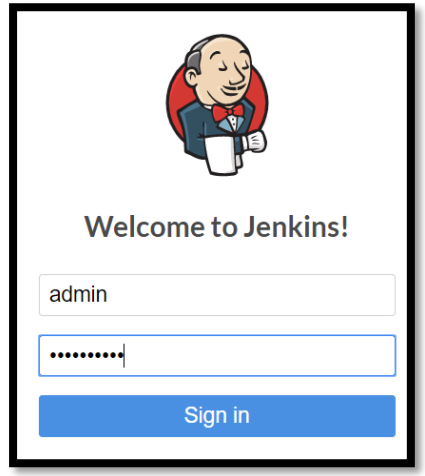
```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/component=jenkins-master" -
l "app.kubernetes.io/instance=cd" -o jsonpath="{.items[0].metadata.name}")

student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ kubectl port-forward $POD_NAME 8080:8080 >> /dev/null &
[3] 2399
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$
```

printf $(kubectl get secret cd-jenkins -o jsonpath=”{.data.jenkins-admin-password}” | base64 — decode);echo

```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ printf $(kubectl get secret cd-jenkins -o jsonpath="{.data.jenkins-admin-password}" | base64 --decode);e
cho
klpHjy74TZ
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$
```





Open web-preview and login as admin with password from last command
click credentials -> Jenkins -> Global Credentials
Click add credentials
select Google Service Account from metadata
Click ok
Click jenkins (top left)
Click new item
enter valkyrie-app
click pipeline
click ok
select pipeline script from SCM
Set SCM to Git
Add the source code repo (find it using gcloud source repos list)

gcloud source repos list

```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ gcloud source repos list
REPO_NAME     PROJECT_ID                     URL
valkyrie-app  qwiklabs-gcp-00-15c467e8f799  https://source.developers.google.com/p/qwiklabs-gcp-00-15c467e8f799/r/valkyrie-app
```

Set credentials to qwiklabs-…

Click save

Change color

sed -i “s/green/orange/g” source/html.go

Update project in Jenkinsfile

sed -i “s/YOUR_PROJECT/$GOOGLE_CLOUD_PROJECT/g” Jenkinsfile

git config — global user.email “you@example.com”

git config — global user.name “student”

git add .

git commit -m “build pipeline init”

git push

```
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ sed -i "s/green/orange/g" source/html.go
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ sed -i "s/YOUR_PROJECT/$GOOGLE_CLOUD_PROJECT/g" Jenkinsfile
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ git config --global user.email "adanikamal@gmail.com"
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ git config --global user.name "student"
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ git add .
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ git commit -m "build pipeline init"
[master 49d83a8] build pipeline init
 4 files changed, 10 insertions(+), 5 deletions(-)
 create mode 100644 Dockerfile
student_00_7544f8e17b13@cloudshell:~/marking/valkyrie-app (qwiklabs-gcp-00-15c467e8f799)$ git push
Counting objects: 8, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 840 bytes | 0 bytes/s, done.
Total 8 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4)
To https://source.developers.google.com/p/qwiklabs-gcp-00-15c467e8f799/r/valkyrie-app
   655d315..49d83a8  master -> master
```

in jenkins click build now on the job

initial build takes a while, just wait

after creating all jobs you will find on jenkins page like this :