Step 1 : Read Dockerfile

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ cat Dockerfile

# Copyright 2016 The Kubernetes Authors.

#

# Licensed under the Apache License, Version 2.0 (the "License");

# you may not use this file except in compliance with the License.

# You may obtain a copy of the License at

#

# http://www.apache.org/licenses/LICENSE-2.0

#

# Unless required by applicable law or agreed to in writing, software

# distributed under the License is distributed on an "AS IS" BASIS,

# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

# See the License for the specific language governing permissions and

# limitations under the License.

FROM golang:1.10.0

RUN go get github.com/codegangsta/negroni \

github.com/gorilla/mux \

github.com/xyproto/simpleredis

WORKDIR /app

ADD ./main.go .

RUN CGO\_ENABLED=0 GOOS=linux go build -o main .

FROM scratch

WORKDIR /app

COPY --from=0 /app/main .

COPY ./public/index.html public/index.html

COPY ./public/script.js public/script.js

COPY ./public/style.css public/style.css

CMD ["/app/main"]

EXPOSE 3000

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$

step 2 : Build Docker image

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ sudo docker build -t aparnatathod/ride:guestboook-go-0.1 .

Step 3 : List the docker images

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ sudo docker image ls

REPOSITORY TAG IMAGE ID CREATED SIZE

aparnatathod/ride guestboook-go-0.1 0f4fa4ee691b 6 minutes ago 9.23MB

<none> <none> 85f543fded72 6 minutes ago 804MB

aparnatathod/ride 0.1 c00f0e5c29aa 19 hours ago 916MB

python 3.6.9 5bf410ee7bb2 2 years ago 913MB

golang 1.10.0 d632bbfe5767 4 years ago 779MB

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$

Above response shows that image “aparnatathod/ride:guestboook-go-0.1” is created

step 4 : Push docker image to docker hub.

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ sudo docker push aparnatathod/ride:guestboook-go-0.1

The push refers to repository [docker.io/aparnatathod/ride]

681d3bdf0d8a: Pushed

452b1bb0a9fd: Pushed

111ae31d66cc: Pushed

e20f2b3f55f9: Pushed

f8646ded36d6: Pushed

guestboook-go-0.1: digest: sha256:5f9decf677bf68753020ce14066c2015395c7d3aacec1311b19843cf36d19ede size: 1356

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$

step 5 : Get the Kubernetes all info

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl get allNAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 14d

[vagrant@master](mailto:vagrant@master):~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$

step 6 : Describe kubernetes nodes info

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl get node -o wide

NAME STATUS ROLES AGE VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE KERNEL-VERSION CONTAINER-RUNTIME

master Ready control-plane 14d v1.25.1 10.0.2.15 <none> Ubuntu 18.04.6 LTS 4.15.0-163-generic containerd://1.5.5

node1 Ready <none> 14d v1.25.1 10.0.2.15 <none> Ubuntu 18.04.6 LTS 4.15.0-163-generic containerd://1.5.5

[vagrant@master](mailto:vagrant@master):~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$

step 7 : Create redis master controller

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl create -f redis-master-controller.json

replicationcontroller/redis-master created

Step 8 :

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl get all

NAME READY STATUS RESTARTS AGE

pod/redis-master-8kwkv 1/1 Running 0 36s

NAME DESIRED CURRENT READY AGE

replicationcontroller/redis-master 1 1 1 36s

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 14d

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$

Step 9 : list the controller

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl get rc

NAME DESIRED CURRENT READY AGE

redis-master 1 1 1 6m12s

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$

Step 10 : The pod is created by replica controller and the Pod is running.

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl get pod redis-master-8kwkv -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

redis-master-8kwkv 1/1 Running 0 3m44s 10.244.1.19 node1 <none> <none>

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$

Step 11 : Create redis master seervice

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl create -f redis-master-service.json

service/redis-master created

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl get all

NAME READY STATUS RESTARTS AGE

pod/redis-master-8kwkv 1/1 Running 0 5m34s

NAME DESIRED CURRENT READY AGE

replicationcontroller/redis-master 1 1 1 5m34s

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 14d

service/redis-master ClusterIP 10.108.9.235 <none> 6379/TCP 9s

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 14d

redis-master ClusterIP 10.108.9.235 <none> 6379/TCP 25s

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$

Step 12 : Create replica controller

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl create -f redis-replica-controller.json

replicationcontroller/redis-replica created

Get the details of the replica controller

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl get rc

NAME DESIRED CURRENT READY AGE

redis-master 1 1 1 11m

redis-replica 2 2 0 13s

Check all the running resources

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl get all

NAME READY STATUS RESTARTS AGE

pod/redis-master-8kwkv 1/1 Running 0 11m

pod/redis-replica-7dhdr 1/1 Running 0 26s

pod/redis-replica-vtswt 1/1 Running 0 26s

NAME DESIRED CURRENT READY AGE

replicationcontroller/redis-master 1 1 1 11m

replicationcontroller/redis-replica 2 2 2 26s

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 14d

service/redis-master ClusterIP 10.108.9.235 <none> 6379/TCP 6m14s

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$

Step 13 : Check the master and replica controller pods are running

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl get pods

NAME READY STATUS RESTARTS AGE

redis-master-8kwkv 1/1 Running 0 17m

redis-replica-7dhdr 1/1 Running 0 6m44s

redis-replica-vtswt 1/1 Running 0 6m44s

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$

As shown above 3 pods are created.

Step 14 : create replica service and check the services running.

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl create -f redis-replica-service.json

service/redis-replica created

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 14d

redis-master ClusterIP 10.108.9.235 <none> 6379/TCP 15m

redis-replica ClusterIP 10.101.85.15 <none> 6379/TCP 12s

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$

Above response displayes three servicse : 1. kubernetes service

2. redis master service

3. redis replica service

Step 15 : create guestbook controller

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl create -f guestbook-controller.json

replicationcontroller/guestbook created

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$

Step 16 : Check the guestbook controller is running.

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl get rc

NAME DESIRED CURRENT READY AGE

guestbook 3 3 3 2m37s

redis-master 1 1 1 128m

redis-replica 2 2 2 116m

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$

Guestbook controller with 3 pods is running.

Step 17 : check all the running pods in default namespaces.

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

guestbook-96ssq 1/1 Running 0 123m 10.244.1.25 node1 <none> <none>

guestbook-v78dk 1/1 Running 0 123m 10.244.1.24 node1 <none> <none>

guestbook-wc5sf 1/1 Running 0 123m 10.244.1.23 node1 <none> <none>

redis-master-8kwkv 1/1 Running 0 4h9m 10.244.1.20 node1 <none> <none>

redis-replica-7dhdr 1/1 Running 0 3h58m 10.244.1.21 node1 <none> <none>

redis-replica-vtswt 1/1 Running 0 3h58m 10.244.1.22 node1 <none> <none>

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go

Result : a single Redis master, two Redis replicas, and three guestbook pods are running.

Step 18 : Create guestbook service with load balancer

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl create -f guestbook-service.json

service/guestbook created

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$ kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

guestbook LoadBalancer 10.106.238.134 <pending> 3000:31588/TCP 40s

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 15d

redis-master ClusterIP 10.108.9.235 <none> 6379/TCP 4h23m

redis-replica ClusterIP 10.101.85.15 <none> 6379/TCP 4h8m

vagrant@master:~/demos/devops-assignment-main-solution/devops-assignment-main/CI/examples-ridecell/guestbook-go$

As kubernetes cluster is created in virtual machines using vagrant and virtual box, so there is no loadbalancer provider is present. Hence theExternal-IP of guestbook loadbalancer is in pending status.