Create a Docker Image and Deploy It to Kubernetes

Course-end Project 1

Description

You are working as a DevOps engineer in an IT firm. You have been asked to create a Redis-based Docker image and deploy it on a Kubernetes cluster.

Background of the problem statement:

Your organization wants to use Redis in a Kubernetes cluster for the data storage and caching purpose. The development team has asked you to create a Redis-based Docker image using a Dockerfile and deploy this image on a Kubernetes cluster.

You have also been asked to publish this image on your organization's Docker Hub account so that other team members can also access this image.

You must use the following:

- Docker CLI: To create the Docker image using a Dockerfile
- Docker Hub: To publish the image
- Kubectl: To deploy the image on a Kubernetes cluster

Following requirements should be met:

- Follow the above-mentioned specifications
- Make sure you create an account on Docker Hub to push the Docker image
- Document the step-by-step process involved in completing this task

1)Create Kubernetes cluster

Run the following commands in all nodes

apt update

apt install docker.io kubeadm kubectl

In master node run

sudo kubeadm init --pod-network-cidr=192.169.0.0/16

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

kubectl apply -f https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-daemonset-k8s.yaml

in worker nodes use join token

kubeadm join 172.31.10.161:6443 --token rytpdx.t6q3nvkdeid87xys \

> --discovery-token-ca-cert-hash sha256:ad78fed5b14e96d89671fc04abb444ff33a011688a7d6ad83411aba4d022df56

```
  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.10.161:6443 --token rytpdx.t6q3nvkdeid87xys \
    --discovery-token-ca-cert-hash sha256:ad78fed5b14e96d89671fc04abb444ff33a011688a7d6ad83411aba4d022df56
root@master:~# mkdir -p $HOME/.kube
root@master:~# sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
root@master:~#
root@master:~# sudo chown $(id -u):$(id -g) $HOME/.kube/config
root@master:~#
root@master:~# kubectl get nodes
NAME      STATUS     ROLES    AGE     VERSION
master    NotReady   master   4m31s   v1.19.4
worker1   NotReady   <none>   21s     v1.19.4
worker2   NotReady   <none>   9s      v1.19.4
root@master:~# [root@controller ~]# curl https://docs.projectcalico.org/manifests/calico.yaml -O
[root@controller: command not found
root@master:~# curl https://docs.projectcalico.org/manifests/calico.yaml -O
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    73  100    73    0     0    223      0 --:--:-- --:--:-- --:--:--   222
root@master:~# kubectl apply -f https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-daemonset.yaml
error: unable to read URL "https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-daemonset.yaml", server reported 404 Not Found, status code=404
root@master:~# kubectl apply -f https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-daemonset-k8s.yaml
serviceaccount/weave-net created
clusterrole.rbac.authorization.k8s.io/weave-net created
clusterrolebinding.rbac.authorization.k8s.io/weave-net created
role.rbac.authorization.k8s.io/weave-net created
rolebinding.rbac.authorization.k8s.io/weave-net created
daemonset.apps/weave-net created
root@master:~# kubectl get nodes
NAME      STATUS     ROLES    AGE   VERSION
master    NotReady   master   18m   v1.19.4
worker1   NotReady   <none>   14m   v1.19.4
worker2   NotReady   <none>   14m   v1.19.4
root@master:~# alias k=kubectl
```

```
  libltng-ust-ctl4 libltng-ust0 liburcu6 linux-aws-5.3-headers-5.3.0-1019
Use 'sudo apt autoremove' to remove them.
Suggested packages:
  debootstrap docker-doc rinse zfs-fuse | zfsutils
The following packages will be REMOVED:
  docker-ce-cli
The following NEW packages will be installed:
  bridge-utils docker.io ubuntu-fan
0 upgraded, 3 newly installed, 1 to remove and 520 not upgraded.
Need to get 30.4 MB of archives.
After this operation, 34.5 MB disk space will be freed.
Do you want to continue? [Y/n] y
Get:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 bridge-utils amd64 1.5-15ubuntu1 [30.1 kB]
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 docker.io amd64 20.10.21-0ubuntu1~18.04.3 [30.3 MB]
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 ubuntu-fan all 0.12.10 [34.7 kB]
Fetched 30.4 MB in 1s (47.8 MB/s)
Preconfiguring packages ...
(Reading database ... 241919 files and directories currently installed.)
Removing docker-ce-cli (5:19.03.11~3-0~ubuntu-bionic) ...
Selecting previously unselected package bridge-utils.
(Reading database ... 241716 files and directories currently installed.)
Preparing to unpack .../bridge-utils_1.5-15ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.5-15ubuntu1) ...
Selecting previously unselected package docker.io.
Preparing to unpack .../docker.io_20.10.21-0ubuntu1~18.04.3_amd64.deb ...
Unpacking docker.io (20.10.21-0ubuntu1~18.04.3) ...
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../ubuntu-fan_0.12.10_all.deb ...
Unpacking ubuntu-fan (0.12.10) ...
Setting up docker.io (20.10.21-0ubuntu1~18.04.3) ...
Setting up bridge-utils (1.5-15ubuntu1) ...
Setting up ubuntu-fan (0.12.10) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu10.41) ...
root@worker1:/home/labsuser# kubeadm join 172.31.10.161:6443 --token rytpdx.t6q3nvkdeid87xys \
>     --discovery-token-ca-cert-hash sha256:ad78fed5b14e96d89671fc04abb444ff33a011688a7d6ad83411aba4d022df56
[preflight] Running pre-flight checks
        [WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Docker cgroup driver. The recommended driver is "systemd". Please follow the guide at https://kubernetes.io/docs/setup/cri/
        [WARNING SystemVerification]: this Docker version is not on the list of validated versions: 20.10.21. Latest validated version: 19.03
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -oyaml'
```

Kubectl get nodes

Vi Dockerfile

FROM redis

ARG key1=value1

docker build -t redis .

```
root@master:~# docker build -t redis .
Sending build context to Docker daemon  50.32MB
Step 1/2 : FROM redis
latest: Pulling from library/redis
52d2b7f179e3: Already exists
689bed60e397: Pull complete
2f34c7846499: Pull complete
723b2c9888ad: Pull complete
16acd9ca1349: Pull complete
29771da5b50b: Pull complete
Digest: sha256:c45b9ac48fde5e7ffc59e785719165511b1327151c392c891c2f552a83446847
Status: Downloaded newer image for redis:latest
 ---> 506734eb5e71
Step 2/2 : ARG key1=value1
 ---> Running in ae0874af6494
Removing intermediate container ae0874af6494
 ---> b8121a5e193e
Successfully built b8121a5e193e
Successfully tagged redis:latest
root@master:~# docker images
REPOSITORY                          TAG        IMAGE ID       CREATED          SIZE
redis                               latest     b8121a5e193e   20 seconds ago   138MB
```

Vi deploy.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: redis
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      bb: redis
  template:
    metadata:
      labels:
        bb: redis
    spec:
      containers:
      - name: redis
        image: redis-test
        imagePullPolicy: Never
```

vi service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: redis-service
  namespace: default
spec:
  type: NodePort
  selector:
    bb: redis
  ports:
  - port: 6379
    targetPort: 6379
    nodePort: 30001
```

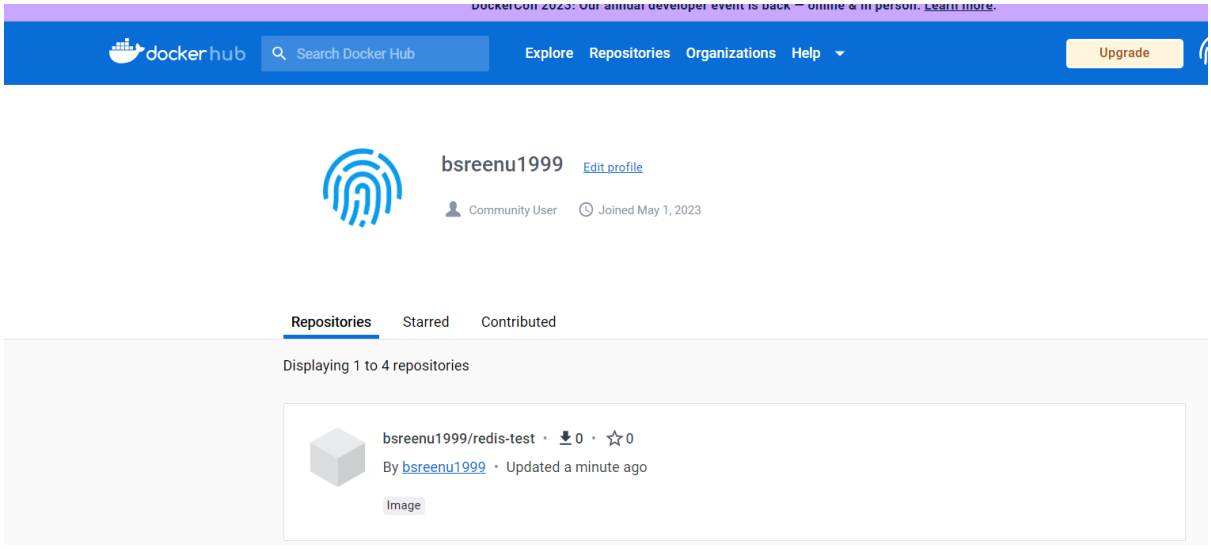
kubectl apply -f deploy.yaml

Kubectl apply -f service.yaml

```
Step 1/2 : FROM redis
 ---> b8121a5e193e
Step 2/2 : ARG key1=value1
 ---> Running in 21fb5a99a71c
Removing intermediate container 21fb5a99a71c
 ---> 67524e8b339c
Successfully built 67524e8b339c
Successfully tagged redis-test:latest
root@master:~# docker images
REPOSITORY                                TAG       IMAGE ID       CREATED          SIZE
redis-test                                latest    67524e8b339c   6 seconds ago    138MB
redis                                     latest    b8121a5e193e   3 minutes ago    138MB
redis                                     <none>    506734eb5e71   3 hours ago      138MB
test                                      2         9540c17e9e4b   4 hours ago      77.8MB
bsreenu1999/ngin                          latest    eea7b3dcba7e   27 hours ago     187MB
nginx                                     latest    eea7b3dcba7e   27 hours ago     187MB
localhost:5000/nginx                      latest    eea7b3dcba7e   27 hours ago     187MB
localhost:50000/nginx                     latest    eea7b3dcba7e   27 hours ago     187MB
registry                                  2         0030ba3d620c   8 days ago       24.1MB
ubuntu                                    latest    01f29b872827   13 days ago      77.8MB
nginx                                     <none>    89da1fb6dcb9   2 weeks ago      187MB
k8s.gcr.io/kube-apiserver                 v1.19.16  8d6534c805c0   21 months ago    119MB
k8s.gcr.io/kube-controller-manager        v1.19.16  a736172e2720   21 months ago    111MB
k8s.gcr.io/kube-proxy                     v1.19.16  8bbb057ceb16   21 months ago    98.9MB
k8s.gcr.io/kube-scheduler                 v1.19.16  7cd6ae6db41e   21 months ago    46.5MB
weaveworks/weave-npc                      latest    690c3345cc9c   2 years ago      39.3MB
weaveworks/weave-kube                     latest    62fea85d6052   2 years ago      89MB
k8s.gcr.io/etcd                           3.4.13-0  0369cf4303ff   2 years ago      253MB
k8s.gcr.io/kube-proxy                     v1.18.8   0fb7201f92d0   3 years ago      117MB
k8s.gcr.io/kube-controller-manager        v1.18.8   6a979351fe5e   3 years ago      162MB
k8s.gcr.io/kube-apiserver                 v1.18.8   92d040a0dca7   3 years ago      173MB
k8s.gcr.io/kube-scheduler                 v1.18.8   6f7135fb47e0   3 years ago      95.3MB
k8s.gcr.io/coredns                        1.7.0     bfe3a36ebd25   3 years ago      45.2MB
k8s.gcr.io/pause                          3.2       80d28bedfe5d   3 years ago      683kB
k8s.gcr.io/coredns                        1.6.7     67da37a9a360   3 years ago      43.8MB
k8s.gcr.io/etcd                           3.4.3-0   303ce5db0e90   3 years ago      288MB
root@master:~# vi deploy.yaml
root@master:~# vi service.yaml
root@master:~# kubectl apply -f deploy.yaml
deployment.apps/redis created
root@master:~# kubectl apply -f service.yaml
service/redis-service created
root@master:~#
```

docker tag redis-test bsreenu1999/redis-test

docker push bsreenu1999/redis-test




```
root@master:~# docker push bsreenu1999/redis-test
Using default tag: latest
The push refers to repository [docker.io/bsreenu1999/redis-test]
6b45f041a0f4: Mounted from library/redis
f9e931d9415a: Mounted from library/redis
ae989b5eb6a6: Mounted from library/redis
c44e5d4e8bf9: Mounted from library/redis
90413ed13ba6: Mounted from library/redis
511780f88f80: Mounted from bsreenu1999/ngin
latest: digest: sha256:a96c34f90214d28ea5cc356b5dedf6990f451267431e7f74f21e39baf8b869d3 size: 1574
root@master:~#
```