DCA Course Project - 1

Project: Create a Docker Image and Deploy It to Kubernetes

Steps :

1. Install Docker

```
$ sudo apt-get remove docker docker-engine docker.io containerd runc
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
docker-compose-plugin
```

2. Setup Kubernetes Cluster

```
#Set Hostname

# In Master Node:
$ sudo hostnamectl set-hostname master.example.com
$ exec bash

# In Worker1 Node:
$ sudo hostnamectl set-hostname worker-node-1.example.com
$ exec bash

# In Worker2 Node:
$ sudo hostnamectl set-hostname worker-node-2.example.com
$ exec bash

# Docker Configuration - Master, Worker1, Worker2

$ sudo mkdir /etc/docker

$ cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"],
"log-driver": "json-file",
"log-opts": {
"max-size": "100m"
},
"storage-driver": "overlay2"
```

```
}
EOF
-----------------------------------------------------------------
$ sudo systemctl enable docker
$ sudo systemctl daemon-reload
$ sudo systemctl restart docker
$ sudo swapoff -a

# Note: Do the above steps in Master, Worker1 and Worker2 nodes

# Master Node initialisation :

$ sudo kubeadm init --pod-network-cidr=192.168.0.0/16

# Copy kubeadm join command

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

# Install Container Network Interface (CNI)

$ kubectl apply -f
https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version |
base64 | tr -d '\n')

# Verification:
$ kubectl get nodes

# Worker Nodes initialization - Worker1 & 2
# DONT COPY AND PASTE: your join command will be different

$ sudo kubeadm join 172.31.59.10:6443 --token lmgp27.x5juohzs4wxmtwdc
\
--discovery-token-ca-cert-hash
sha256:a059ff766ec0801fb80716f4e8614008092a087ccf8f5b97097c26d4213a77
12
#Note: In case you need to find your unique token, run the command
```

DCA Course Project - 1

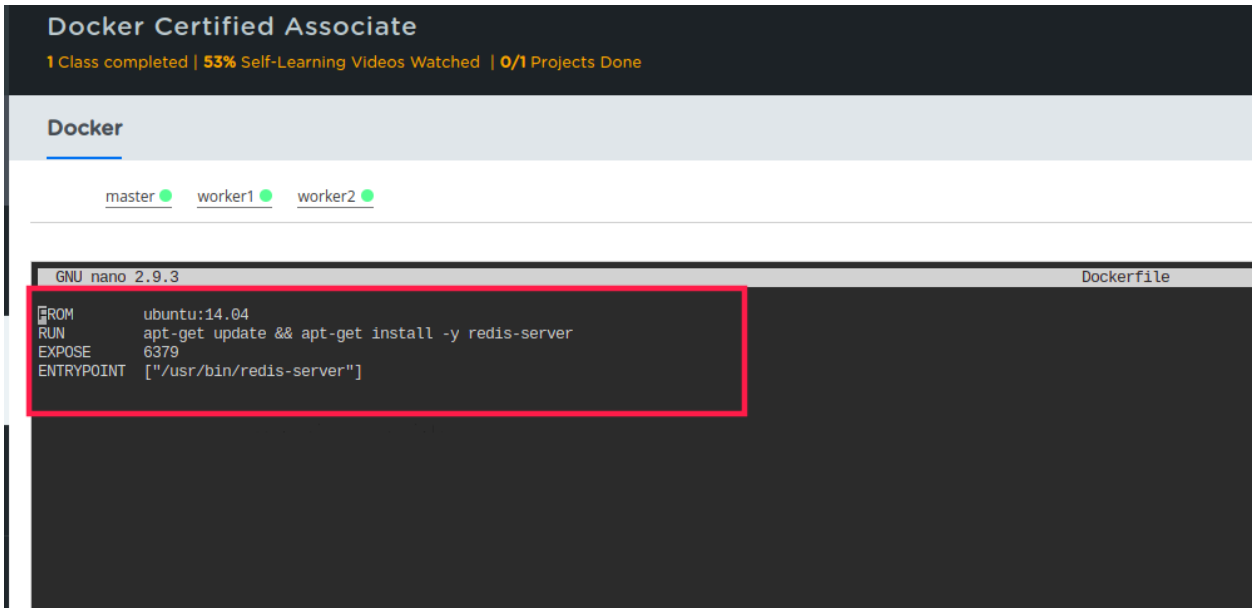
```
$ sudo kubeadm token create --print-join-command
```

3. Create a Dockerfile for redis image

```
$ nano Dockerfile

# Paste the below commands in Dockerfile

FROM        ubuntu:14.04
RUN         apt-get update && apt-get install -y redis-server
EXPOSE      6379
ENTRYPOINT  ["/usr/bin/redis-server"]
```



4. Build the Dockerfile to create a redis image

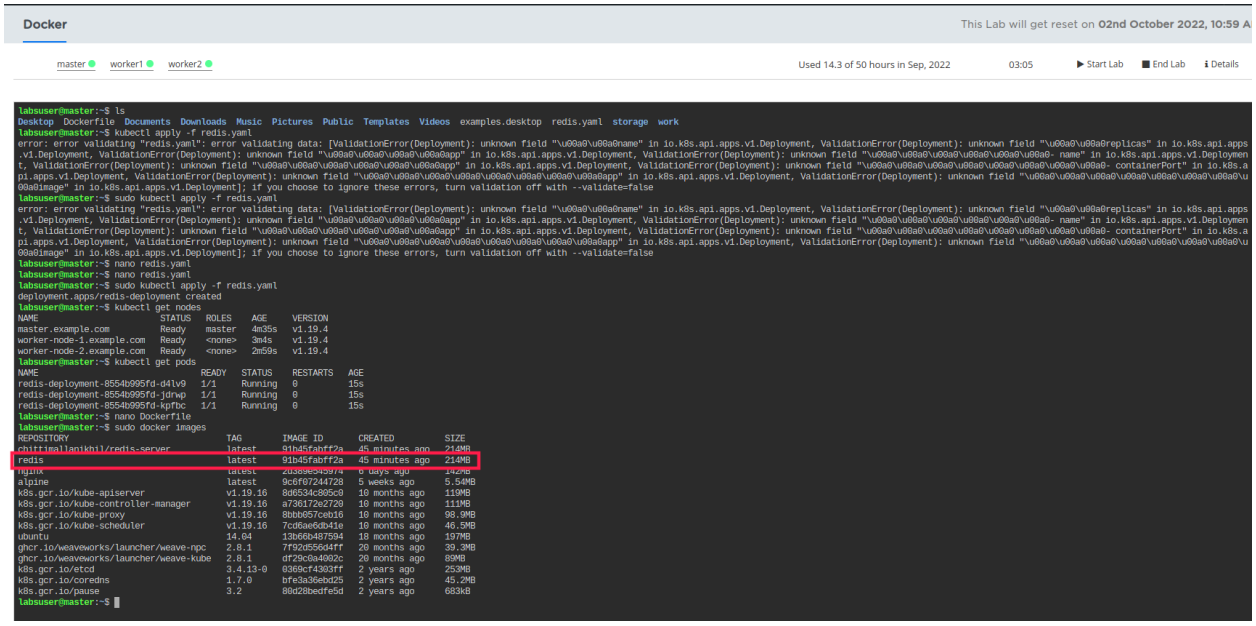
```
$ sudo docker build -t redis .
```

5. Verifying the image is created or not

```
$ sudo docker images

# you can see the redis image created there.
```

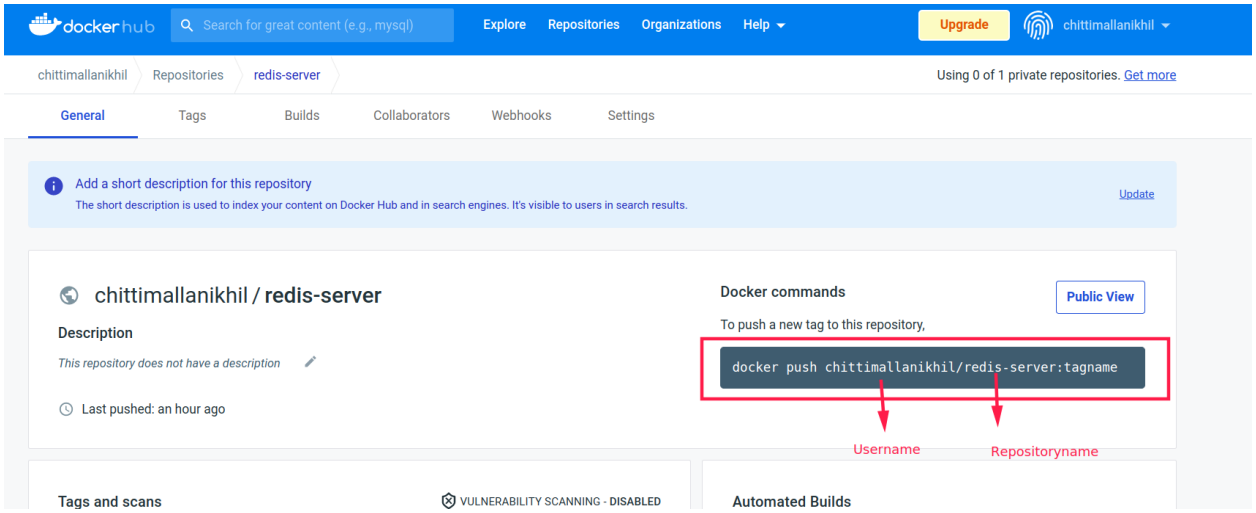
DCA Course Project - 1



6. Tag and push the Image to DockerHub

```
# Need to create a DockerHub account by using the link
https://hub.docker.com/

$ sudo docker tag redis:latest <yourUserName/Repositoryname>:tagname
$ sudo docker push <yourUserName/Repositoryname>:tagname
```



DCA Course Project - 1

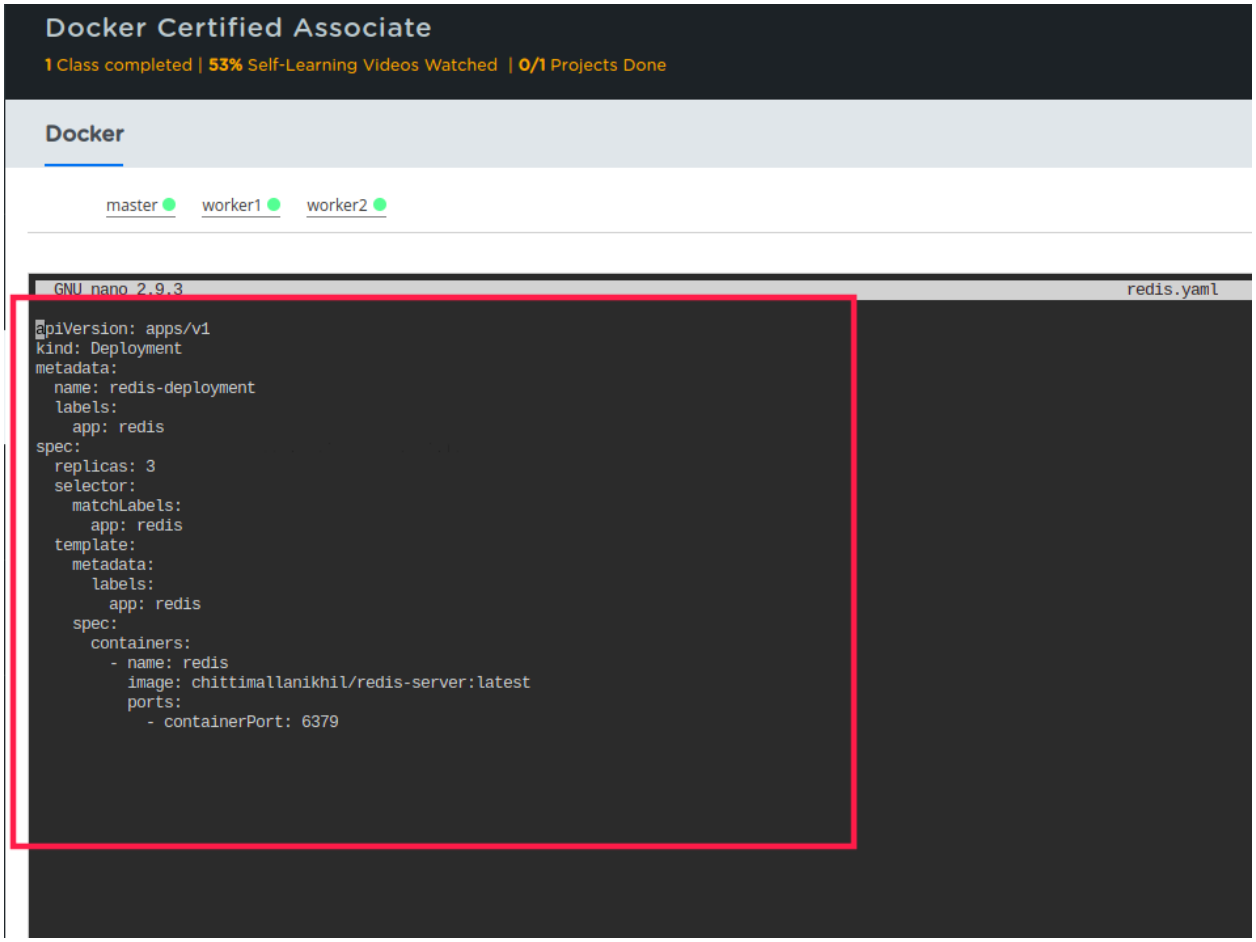
7. Create a yaml file for image deployment in kubernetes cluster

```
$ nano redis.yaml

# Note : paste the below text into yaml file and change the image
name in it with your docker hub account details.

apiVersion: apps/v1
kind: Deployment
metadata:
  name: redis-deployment
  labels:
    app: redis
spec:
  replicas: 3
  selector:
    matchLabels:
      app: redis
  template:
    metadata:
      labels:
        app: redis
    spec:
      containers:
        - name: redis
          image: chittimallanikhil/redis-server:latest
          ports:
            - containerPort: 6379
```

DCA Course Project - 1



8. Deploy the image in kubernetes cluster

```
$ sudo kubectl apply -f redis.yaml

# you can see deployment created and verify it.

$ sudo kubectl get pods

# you can see the pods created

$ sudo kubectl get deployment

# you can see the deployment created
```

DCA Course Project - 1

Output :