Developers Cloud Playground

Minikube Kubernetes with Docker driver in RedHat (RockyLinux) VDI

Sudhakar Krishnamachari - August 2022

Initial Installations Required:

1. Virtualbox : https://www.virtualbox.org/wiki/Downloads
2. RedHat Linux Or Rocky Linux 8.6 : (DVD image 10+ GB) :
   https://download.rockylinux.org/pub/rocky/8/isos/x86_64/Rocky-8.6-x86_64-dvd1.iso

Install / configure RHEL / Rocky on VBox: (No LIVE CD)

* VDI Disk >= 50GB ; RAM >= 8GB ; Processors >= 4 cpus
* Base server with GUI install for software selection (no additional features for now)

(Presume host laptop config of free HDD >= 250 GB; RAM >= 16GB ; Processors >= 8)

User creation in the initial steps, wizard/GUI in virtual box linux installation run, should carry advanced config to include the user as admin “wheel” and “docker” usergroups for required permissions to flow through.

Gotcha’s : (maybe few more…)

- The screen resolution stays at 800x600 despite change through desktop >> Display settings changes on reboot.
- Wired connection needs to be switched on every time after reboot

Install Chrome Browser from firefox google search and install through Software Installer of the downloaded rpm automatically : https://www.google.com/chrome/ Choose rpm in the dialog; make chrome default browser . After every reboot check on browser if internet works

Linux Installations: Mostly on terminal / bash shell

1. Docker engine:

Install and check steps:

```
$ sudo dnf update --nobest -y
$ reboot                          ( and log back again )
```

Uninstall podman / buildah if it exists >

```
$ sudo yum erase podman buildah
$ sudo dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo
$ sudo yum install docker-ce
$ systemctl start docker.service
$ docker run hello-world
```

(needs root permissions (viz: wheel usergroup); though option for rootless docker on rhel9.0+ exists) ; validates working docker engine

2. Minikube kubernetes :

$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-latest.x86_64.rpm
$ sudo rpm -Uvh minikube-latest.x86_64.rpm

Finally a nice presentation of the blank kubernetes cluster and deployments; should automatically open in chrome browser : Open a new terminal tab and run this:
$ minikube dashboard

Start the kubernetes minikube system: Back to terminal tab 01:
$ minikube start

install kubectl through minikube:
$ minikube kubectl -- get po -A
$ alias kubectl="minikube kubectl –" *(this may be required in each terminal tab opened for the session; else all command line action will have to be with "minikube kubectl --")*

First Kubernetes Deployment and Test:

$ kubectl create deployment hello-minikube –image=k8s.gcr.io/echoserver:1.4
$ kubectl expose deployment hello-minikube --type=NodePort –port=8080
$ kubectl port-forward deployment/hello-minikube 8080:8080
check on browser : http://localhost:8080

Logs access:
$ minikube kubectl -- logs deployment.apps/hello-minikube

Tomcat-Sample Webapp Deployment on Minikube :

1. Create directory hierarchy your choice under:
/home/{yourloginid}
> techworks
> cloud
> minikube
> projects
> sample-tomcatwebapp-project
> kube

2. Now the docker and minikube metadata:
Open gedit from Activities or terminal:

3. change directory to "sample-project"
Create minikube: deployment.yaml *(replace "." with " " space to save the yaml)*

apiVersion:.apps/v1
kind:.Deployment
metadata:
..name:.tomcat-sample
..labels:

```
....name:.tomcat-sample
spec:
..replicas:.3
..selector:
....matchLabels:
......name:.tomcat-sample
..template:
....metadata:
......labels:
........name:.tomcat-sample
....spec:
......containers:
........-.name:.tomcat-sample
..........image:.tomcat9jdk11/sample
..........imagePullPolicy:.Never
..........ports:
............-.containerPort:.3333
```

*In eventual state of fully developed Kubernetes design, will have **services, config-map, storage and other yaml files** als. For the basics this is adequate for now.*

Change directory to app : viz: cd app :
Create file: Dockerfile :in app folder

```
FROM tomcat:9.0
LABEL maintainer="krishnamachari.sudhakar@gmail.com"
ADD sample.war /usr/local/tomcat/webapps/
RUN sed -i 's/port="8080"/port="3333"/' ${CATALINA_HOME}/conf/server.xml
CMD ["catalina.sh", "run"]
```

Add a sample.war of your own or from internet. Just needs index.html; WEB-INF / META-INF and additional files as your choice, including a hello world servelet, JSP et als.

Place the sample.war in app folder alongside Dockerfile

build the docker image in minikube: *(copy and paste including the . in the end)*

```
$ minikube image build -t tomcat9jdk11/sample .
$ kubectl logs deployment.apps/tomcat-sample  (check logs..)
```

Change directory to “sample-tomcatwebapp-docker/kube” viz: cd..
run the kubernetes deployment:

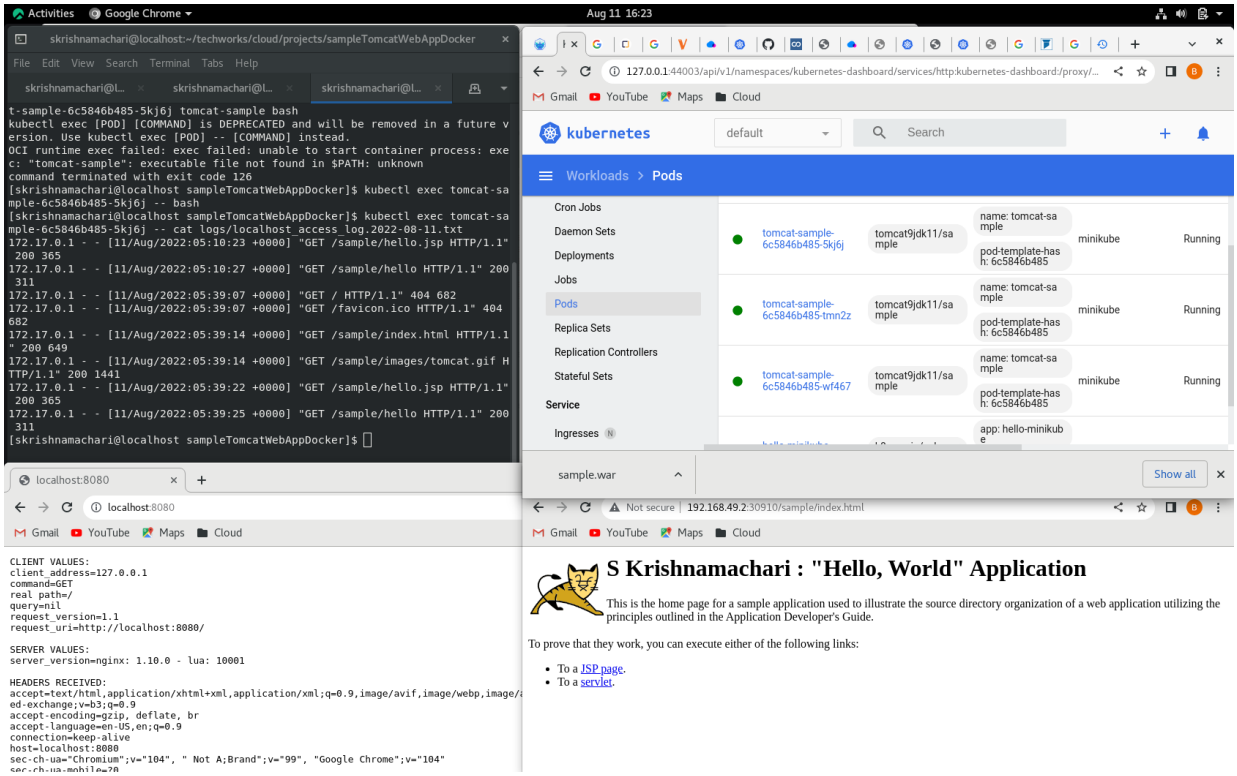
```
$ kubectl apply -f deployment.yaml
```

check in minikube dashboard if already open in browser and running (the terminal tab 02 is still running)
check through Deployment, Pods, ReplicaSets, the 3 containers launched ...

```
$ kubectl logs deployment.apps/tomcat-sample (check logs..)
```

Run kubernetes service to expose the running tomcat to the cluster

```
$ kubectl expose deployment tomcat-sample --type=NodePort --name=tomcat-sample-service
```



Open the service on browser: The fully enabled, deployment ..!!

$ minikube service tomcat-sample-service

this will open on a cluster IP with a random port assigned as load balancer kube-proxy access. Navigate to the two links in the index.html page: hello servlet and JSP page

Logs access per pod:

$ kubectl logs deployment.apps/tomcat-sample-xxxxxxxxxx

get the name of the pods from the minikube dashboard or from:

$ kubectl get pods

Check logs on all 3 pods .. similar and running tomcat webapp logging shown

Access the containers bash shell to look at internal tomcat logs

run these through all 3 pods for logging load balanced in all 3 pods

$ kubectl exec tomcat-sample-xxxxxxxx -- ls logs/*

(list all log files)

$ kubectl exec tomcat-sample-xxxxxxx -- cat logs/localhost_access_log.{yyyy-mm-dd}.txt

(put the right log file name…)

Deprecated command line:

$ kubectl exec -it tomcat-sample-xxxxxxxxxxx bash

should provide the container prompt :

root@tomcat-sample-xxxxxxxxx:/usr/local/tomcat#

ls logs/* (list all log files)

cat logs/localhost_access_log.{yyyy-mm-dd).txt

exit

References:

Basic:
https://www.backblaze.com/blog/vm-vs-containers/

https://docs.docker.com/get-started/
https://minikube.sigs.k8s.io/docs/start/

Beyond Basic:
https://levelup.gitconnected.com/two-easy-ways-to-use-local-docker-images-in-minikube-cd4dcb1a5379

https://kubernetes.io/docs/tasks/access-application-cluster/service-access-application-cluster/
https://containerjournal.com/editorial-calendar/best-of-2021/kubernetes-pods-vs-deployments/
https://betterprogramming.pub/k8s-a-closer-look-at-kube-proxy-372c4e8b090
https://capstonec.com/2019/12/16/getting-tomcat-logs-from-kubernetes-pods/
https://techvedika.com/docker-security-container-security-tools/
https://kubernetes.io/docs/concepts/services-networking/_print/

Key Components: