K8S DEPLOYMENT

SYSTEM ON DOCKER DESKTOP

Deployment folder: D:\development\K8sDockerDesktop\deployment

Github: https://github.com/BaruchiHalamish20/K8sDockerDesktop-.git

keubctl

Version:

```
$ kubectl version --output=yaml
clientVersion:
  buildDate: "2022-11-09T13:36:36Z"
  compiler: gc
  gitCommit: 872a965c6c6526caa949f0c6ac028ef7aff3fb78
  gitTreeState: clean
  gitVersion: v1.25.4
  goVersion: go1.19.3
  major: "1"
  minor: "25"
  platform: windows/amd64
kustomizeVersion: v4.5.7
serverVersion:
  buildDate: "2022-11-09T13:29:58Z"
  compiler: gc
  gitCommit: 872a965c6c6526caa949f0c6ac028ef7aff3fb78
  gitTreeState: clean
  gitVersion: v1.25.4
  goVersion: go1.19.3
  major: "1"
  minor: "25"
  platform: linux/amd64
```

configuration

```
$ kubectl config get-contexts
CURRENT   NAME                          CLUSTER                       AUTHINFO
NAMESPACE
```

docker-desktop docker-desktop docker-desktop

gke_development-371512_me-west1_development-cluster gke_development-371512_me-west1_development-cluster gke_development-371512_me-west1_development-cluster

* gke_grafana7_me-west1_grafana7-cluster gke_grafana7_me-west1_grafana7-cluster gke_grafana7_me-west1_grafana7-cluster kube-system

kubectl config use-context docker-desktop

Architecture

K8sUpdated.docx

Section: Architecture

Nginx deployment

deployment + service

Deployment

```
$ cat kubefiles/nginx-depl.yaml
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: apps/v1
kind: Deployment
metadata:
 labels:
  app: nginx
 name: nginx-deployment
 namespace: default
spec:
 replicas: 2
 revisionHistoryLimit: 10
 selector:
  matchLabels:
   app: nginx
 strategy:
  rollingUpdate:
   maxSurge: 25%
   maxUnavailable: 25%
  type: RollingUpdate
 template:
```

```
  metadata:
    labels:
      app: nginx
  spec:
    containers:
    - image: nginx:1.16
      name: nginx
      ports:
      - containerPort: 8080
```

```
Baruchi@BARUCHI-PC MINGW64 /d/development/K8sDockerDesktop (master)
$ k apply -f kubefiles/nginx-depl.yaml
$ k describe pod nginx-deployment-78cc6468fb-bzfzb
Name:             nginx-deployment-78cc6468fb-bzfzb
Namespace:        default
Priority:         0
Service Account:  default
Node:             docker-desktop/192.168.65.4
Start Time:       Thu, 02 Mar 2023 16:39:10 +0200
Labels:           app=nginx
                  pod-template-hash=78cc6468fb
Annotations:      <none>
Status:           Running
IP:               10.1.0.23
IPs:
  IP:           10.1.0.23
Controlled By:  ReplicaSet/nginx-deployment-78cc6468fb
Containers:
  nginx:
    Container ID:   docker://de0c7ed88ac6f3e1700e0188f4bfd136ef1e5836b6183608d1f542a4a57349f7
    Image:          nginx:1.16
    Image ID:       docker-pullable://nginx@sha256:d20aa6d1cae56fd17cd458f4807e0de462caf2336f0b70b5eeb69fcaaf30dd9c
    Port:           8080/TCP
    Host Port:      0/TCP
```

Service

```
$ cat kubefiles/nginx-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080


Baruchi@BARUCHI-PC MINGW64 /d/development/K8sDockerDesktop (master)
$ k apply -f kubefiles/nginx-service.yaml
$ Kubectl describe service nginx-service
Name:              nginx-service
Namespace:         default
Labels:            <none>
Annotations:       <none>
Selector:          app=nginx
Type:              ClusterIP
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                10.96.101.192
IPs:               10.96.101.192
Port:              <unset>  80/TCP
TargetPort:        8080/TCP
Endpoints:         10.1.0.23:8080,10.1.0.24:8080
Session Affinity:  None
Events:            <none>
```

MONGO DB

General

From dockerhub: https://hub.docker.com/_/mongo

Port: 27017

Username: using env.var MONGO_INITDB_ROOT_USERNAME=username

Passowrd: using env.var MONGO_INITDB_ROOT_PASSWORD=password

Goos reference:

https://devopscube.com/deploy-mongodb-kubernetes/

DEPLOYMENT ORDER: SECRET than MONGODB

SECRET

Steps:

1. Base64
2. Adding secret deployment using base 64

Base64

$ echo -n "username" | base64

dXNlcm5hbWU=

$ echo -n "password" | base64

cGFzc3dvcmQ=

Secret deployment

kubefiles/mongodb-secret.yaml

```
Cat $ cat kubefiles/mongodb-secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: mongodb-secret
type: Opaque
data:
  mongodb-root-username: dXNlcm5hbWU=
  mongodb-root-password: cGFzc3dvcmQ=
```

$ k apply -f kubefiles/mongodb-secret.yaml

secret/mongodb-secret created

```
Baruchi@BARUCHI-PC MINGW64 /d/development/K8sDockerDesktop (master)
$ k get secret
NAME           TYPE     DATA   AGE
mongodb-secret   Opaque   2      33s
```

MONGODB Persistent volume PV

```
$ k create -f kubefiles/mongodb-pvc.yaml
persistentvolumeclaim/pvc created

Baruchi@BARUCHI-PC MINGW64 /d/development/K8sDockerDesktop (master)
$ cat kubefiles/mongodb-pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: pvc
spec:
 storageClassName: ""
 accessModes:
  - ReadWriteOnce
 volumeName: pv
 resources:
  requests:
   storage: 1Gi
```

Mongodb deployment

File

```
Baruchi@BARUCHI-PC MINGW64 /d/development/K8sDockerDesktop (master)
$ cat  kubefiles/mongodb-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
 creationTimestamp: null
 labels:
  app: mongodb
 name: mongodb
spec:
```

```
replicas: 1
selector:
  matchLabels:
    app: mongodb
strategy: {}
template:
  metadata:
    creationTimestamp: null
    labels:
      app: mongodb
  spec:
    containers:
    - image: mongo
      name: mongodb
      args: ["--dbpath","/data/db"]
      livenessProbe:
        exec:
          command:
            - mongo
            - --disableImplicitSessions
            - --eval
            - "db.adminCommand('ping')"
        initialDelaySeconds: 30
        periodSeconds: 10
        timeoutSeconds: 5
        successThreshold: 1
        failureThreshold: 6
      readinessProbe:
        exec:
          command:
            - mongo
            - --disableImplicitSessions
            - --eval
            - "db.adminCommand('ping')"
        initialDelaySeconds: 30
        periodSeconds: 10
        timeoutSeconds: 5
        successThreshold: 1
```

```
      failureThreshold: 6
    env:
    - name: MONGO_INITDB_ROOT_USERNAME
      valueFrom:
        secretKeyRef:
          name: mongodb-secret
          key: mongo-root-username
    - name: MONGO_INITDB_ROOT_PASSWORD
      valueFrom:
        secretKeyRef:
          name: mongodb-secret
          key: mongo-root-password
    volumeMounts:
    - name: "mongo-data-dir"
      mountPath: "/data/db"
  volumes:
  - name: "mongo-data-dir"
    persistentVolumeClaim:
      claimName: "pvc"
```

Mongodb internal service

Adding service to mongodb-deployment.yaml

```
---
 apiVersion: v1
kind: Service
metadata:
  name: mongodb-service
spec:
  selector:
    app: mongodb
  ports:
    - protocol: TCP
      port: 27017
      targetPort: 27017
```

Monogo-express config map

Monogo-express config map

/d/development/K8sDockerDesktop/kubefiles (master)

$ cat mongodb-configMap.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: mongodb-configmap
data:
  database_url: mongodb-service
```

$ kubectl get configmap mongodb-configmap -o json | jq -r '.data | to_entries[] | .key + "=" + .value'

database_url=mongodb-service

Mongo-express deployment

Same as mongodb, service is external (spec -> type: Loadbalancer)

File: https://github.com/BaruchiHalamish20/K8sDockerDesktop-/blob/master/kubefiles/mongo-express.yaml

Mongo-express external service

Type: Serice

Spec -> type: LoadBalancer

```
apiVersion: v1
kind: Service
metadata:
  name: mongodb-express-service
spec:
  selector:
    app: mongo-express
```

```
type: LoadBalancer
ports:
 - protocol: TCP
   port: 8081
   targetPort: 8081
   nodePort: 30000
```

Kubernetes-Dashboard

Url:

Command:

kubectl apply –f https://raw.githubusercontent.com/kubernetes/dashboard/v2.2.0/aio/deploy/recommended.yaml

Enable skip-login by adding path:

kubectl patch deployment kubernetes-dashboard -n kubernetes-dashboard --type 'json' -p '[{"op": "add", "path": "/spec/template/spec/containers/0/args/-", "value": "--enable-skip-login"}]'

All elements

```
$ k get all -n kubernetes-dashboard  -o wide
NAME                                        READY   STATUS    RESTARTS   AGE    IP          NODE             NOMINATED NODE   READINESS GATES
pod/dashboard-metrics-scraper-7cc7856cfb-h2rjz   1/1     Running   0          4m27s  10.1.0.65   docker-desktop   <none>           <none>
pod/kubernetes-dashboard-566cc9985c-bsf5c        1/1     Running   0          3m19s  10.1.0.66   docker-desktop   <none>           <none>

NAME                                TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE    SELECTOR
service/dashboard-metrics-scraper   ClusterIP   10.109.109.170   <none>        8000/TCP   4m28s  k8s-app=dashboard-metrics-scraper
service/kubernetes-dashboard        ClusterIP   10.96.201.117    <none>        443/TCP    4m29s  k8s-app=kubernetes-dashboard
```

Comment : Internal service !! - will expose it using ingress

```
NAME                                   READY   UP-TO-DATE   AVAILABLE   AGE    CONTAINERS                IMAGES                                SELECTOR
deployment.apps/dashboard-metrics-scraper   1/1     1            1           4m27s  dashboard-metrics-scraper   kubernetesui/metrics-scraper:v1.0.6   k8s-app=dashboard-metrics-scraper
deployment.apps/kubernetes-dashboard        1/1     1            1           4m28s  kubernetes-dashboard        kubernetesui/dashboard:v2.2.0         k8s-app=kubernetes-dashboard

NAME                                              DESIRED   CURRENT   READY   AGE    CONTAINERS                IMAGES                                SELECTOR
replicaset.apps/dashboard-metrics-scraper-7cc7856cfb   1         1         1       4m27s  dashboard-metrics-scraper   kubernetesui/metrics-scraper:v1.0.6   k8s-app=dashboard-metrics-scraper,pod-template-hash=7cc7856c
```

fb

replicaset.apps/kubernetes-dashboard-54fdbcdf66 0 0 0 4m28s kubernetes-dashboard kubernetesui/dashboard:v2.2.0 k8s-app=kubernetes-dashboard,pod-template-hash=54fdbcdf66

replicaset.apps/kubernetes-dashboard-566cc9985c 1 1 1 3m19s kubernetes-dashboard kubernetesui/dashboard:v2.2.0 k8s-app=kubernetes-dashboard,pod-template-hash=566cc9985c

Ingress deployment

Ingress controller

Url: https://kubernetes.github.io/ingress-nginx/deploy/#quick-start

Command:

kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.6.4/deploy/static/provider/cloud/deploy.yaml

Ingress deployment

Use case : expose dashboard

To implement dashboard pods follow: https://andrewlock.net/running-kubernetes-and-the-dashboard-with-docker-desktop/

kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.2.0/aio/deploy/recommended.yaml

Adding dashboard pods:

```
$ k get all -n kubernetes-dashboard
NAME                                        READY   STATUS    RESTARTS   AGE
pod/dashboard-metrics-scraper-7cc7856cfb-h2rjz   1/1     Running   0          3h40m
pod/kubernetes-dashboard-566cc9985c-bsf5c        1/1     Running   0          3h39m

NAME                                TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
service/dashboard-metrics-scraper   ClusterIP   10.109.109.170  <none>        8000/TCP   3h40m
service/kubernetes-dashboard        ClusterIP   10.96.201.117   <none>        443/TCP    3h40m

NAME                                        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/dashboard-metrics-scraper   1/1     1            1           3h40m
deployment.apps/kubernetes-dashboard        1/1     1            1           3h40m

NAME                                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/dashboard-metrics-scraper-7cc7856cfb  1         1         1       3h40m
replicaset.apps/kubernetes-dashboard-54fdbcdf66       0         0         0       3h40m
replicaset.apps/kubernetes-dashboard-566cc9985c       1         1         1       3h39m
```

Deploy dashboard-ingress:

```
$ cat  kubefiles/dashboard-ingress.yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
 name: dashboard-ingress
 namespace: kubernetes-dashboard
 annotations:
  nginx.ingress.kubernetes.io/rewrite-target: /
  nginx.ingress.kubernetes.io/backend-protocol: "HTTPS"
  nginx.ingress.kubernetes.io/ssl-passthrough: "true"
  kubernetes.io/ingress.class: "nginx"
  nginx.ingress.kubernetes.io/configuration-snippet: |-
   proxy_ssl_server_name on;
   proxy_ssl_name $host;
spec:
 ingressClassName: nginx
 rules:
  - host: dashboard.com
    http:
     paths:
     - backend:
        service:
         name: kubernetes-dashboard
         port:
          number: 443
       path: /
       pathType: ImplementationSpecific
```

$ tail -1 /c/Windows/System32/drivers/etc/hosts

127.0.0.1 dashboard.com

Debug ingress:

k logs ingress-nginx-controller-6b94c75599-dc6sr -n ingress-nginx -f

Permissions

Url: https://github.com/kubernetes/dashboard/blob/master/docs/user/access-control/creating-sample-user.md

Assing RBAC to admin user:

$ k apply -f kubefiles/dashboard-admin.yaml

serviceaccount/admin-user created

clusterrolebinding.rbac.authorization.k8s.io/admin-user created

+ SKIP

Generate token DID NOT HELP LOGIN

$ kubectl -n kubernetes-dashboard create token admin-user

eyJhbGciOiJSUzI1NiIsImtpZCI6IlhnWDRPNUtaQXhXSmkxTUxPY0t2N1JfdFgza1NLUUlrN2ctc25Ic0VWcWMifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiXSwiZXhwIjoxNjc4Mjg3MDYxLCJpYXQiOjE2NzgyODM0NjEsImlzcy

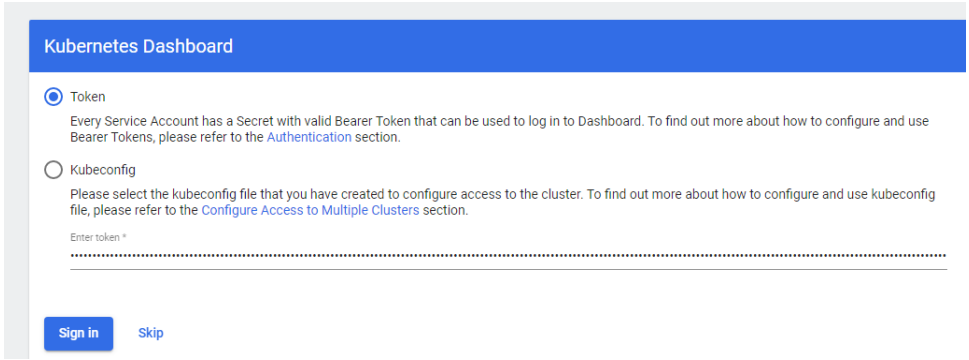
I6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJrdWJlcm5ldGVzLWRhc2hib2FyZCIsInNlcnZpY2VhY2NvdW50Ijp7Im5hbWUiOiJhZG1pbi11c2VyIiwidWlkIjoiNDEwMzdiM2EtYmJkNS00ODh

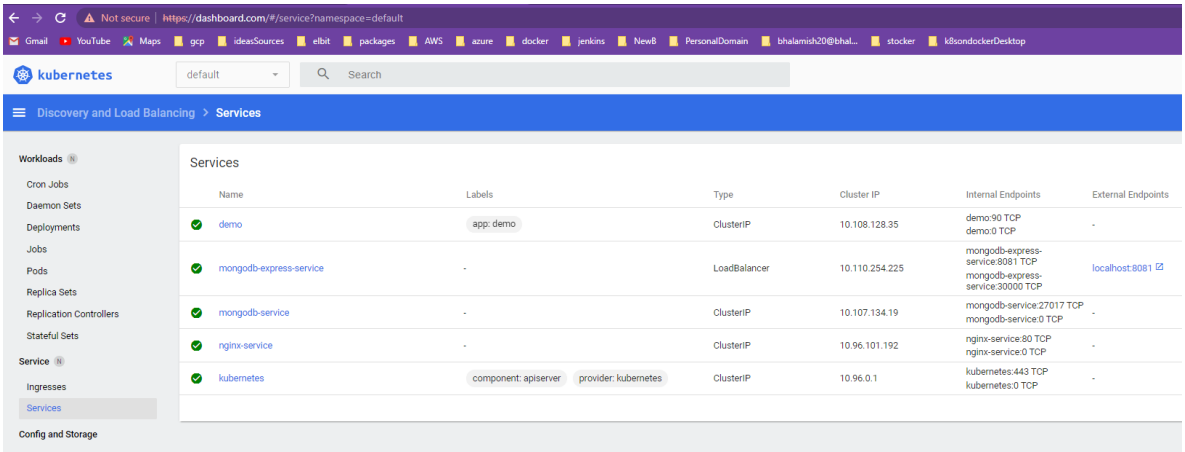
hLThhNDItZDZiODNkZWNiN2M5In19LCJuYmYiOjE2NzgyODM0NjEsInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDprdWJlcm5ldGVzLWRhc2hib2FyZDphZG1pbi11c2VyIn0.GMJeOVCcvrnTkYp23uT-d5kCHanWXj9VhqYjThII6WcoMCwnnES-4S2nKyz1mk3AmXiENWywtLiesAId

V4GQqTDOyjFHu2y5pzruV_fJTqC4akNZraykrjKf-S3Q8xwtrI_eQ8nHAmM3vpRlKJilxBx4_jm7M-xCO0X0GtuKp945OpBKfpS9VWFoqnQ9kc0eRy-PCEjvpSBeoS5E6UGz2MdKiPUMq6nlFx7E-g7yxLR_fjfJNv7G3N5HPGLkq0d1ZBxW9YRnUzwqHZHBWzB05T4B-j1aH00p4801b9pwy

QTDxLCik2Ovb1fX3ZxveSqd6aTG4_0I64a2ZoEN4nGLVg

https://dashboard.com/#/login





==--,][

Persistant Volume

Below are the fils for pv , claim and 2 pods:

$ ls -ltr

total 4

-rw-r--r-- 1 Baruchi 197121 166 Mar 12 14:46 pv.yaml

-rw-r--r-- 1 Baruchi 197121 155 Mar 12 14:47 pvc.yaml

-rw-r--r-- 1 Baruchi 197121 285 Mar 12 15:06 use_pvc_second.yaml

-rw-r--r-- 1 Baruchi 197121 271 Mar 12 15:07 use_pvc.yaml

Tests:

Create the first file on "/data"

$ k exec -it pod/use-pvc bash

root@use-pvc:/# cd /data

root@use-pvc:/data# touch created-by-use-pvc-pod

root@use-pvc:/data# ls -ltr

total 0

-rw-r--r-- 1 root root 0 Mar 12 13:04 created-by-use-pvc-pod

Create the second on /data and verify that the first one exist:

$ k exec -it use-pvc-second bash

root@use-pvc-second:/# ls /data

created-by-use-pvc-pod

root@use-pvc-second:/# touch /data/created-by-second-pod

Verify that the second pod can access the file created by the first pod:

$ k exec -it use-pvc bash

root@use-pvc:/# ls -l /data

total 0

-rw-r--r-- 1 root root 0 Mar 12 13:09 created-by-second-pod

-rw-r--r-- 1 root root 0 Mar 12 13:04 created-by-use-pvc-pod

OPEN SSL / TLS

https://www.stechies.com/installing-openssl-windows-10-11/