Kubernetes with Docker for Desktop

- Enable Kubernetes
- Install the Dashboard
- Install Sample Pod
- More Advances Examples
    - Node Storage
    - Minecraft Server with Resource Declarations
- Rest Cluster/Start Over
- References

Enable Kubernetes

From Docker Preferences, enable Kubernetes.

blocked URL

Check to see if it is running by issuing the following command:

> kubectl get nodes

```
    NAME                 STATUS   ROLES    AGE   VERSION
docker-for-desktop   Ready    master   5h    v1.10.3
```

Install the Dashboard

> kubectl create -f https://raw.githubusercontent.com/kubernetes/dashboard/master/aio/deploy/recommended/kubernetes-dashboard.yaml

Verify that it is installed:

> kubectl get pods --all-namespaces -o wide

```
    NAMESPACE     NAME                                         READY   STATUS    RESTARTS   AGE
IP             NODE
docker        compose-74649b4db6-zpb5t                     1/1     Running   0          5h    10.1.0.3
docker-for-desktop
docker        compose-api-8477889868-jfzph                 1/1     Running   0          5h    192.168.65.3
docker-for-desktop
kube-system   etcd-docker-for-desktop                      1/1     Running   0          5h    192.168.65.3
docker-for-desktop
kube-system   kube-apiserver-docker-for-desktop            1/1     Running   0          5h    192.168.65.3
docker-for-desktop
kube-system   kube-controller-manager-docker-for-desktop   1/1     Running   0          5h    192.168.65.3
docker-for-desktop
kube-system   kube-dns-86f4d74b45-txr8w                    3/3     Running   0          5h    10.1.0.2
docker-for-desktop
kube-system   kube-proxy-q5vrd                             1/1     Running   0          5h    192.168.65.3
docker-for-desktop
kube-system   kube-scheduler-docker-for-desktop            1/1     Running   0          5h    192.168.65.3
docker-for-desktop
kube-system   kubernetes-XXX                                       1/1     Running   0
50s   10.1.0.4       docker-for-desktop
```

Create a dashboard admin user:

vi dashboard-adminuser.yaml

```
    apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kube-system
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kube-system
```

> kubectl apply -f dashboard-adminuser.yaml

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

Get the Token for the created user

> kubectl -n kube-system describe secret $(kubectl -n kube-system get secret | grep admin-user | awk '{print $1}')

```
    Name:         admin-user-token-shqsn
Namespace:    kube-system
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: admin-user
              kubernetes.io/service-account.uid: 8ab8e997-1069-11e9-9ec0-025000000001

Type:  kubernetes.io/service-account-token

Data
====
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6IiJ9.
eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrdWJlLXN5
c3RlbSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJhZG1pbi11c2VyLXRva2VuLXNocXNuIiwia3ViZXJuZXRl
cy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6ImFkbWluLXVzZXIiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50
L3NlcnZpY2UtYWNjb3VudC51aWQiOiI4YWI4ZTk5Ny0xMDY5LTExZTktOWVjMC0wMjUwMDAwMDAwMDEiLCJzdWIiOiJzeXN0ZW06c2VydmljZWFj
Y291bnQ6a3ViZS1zeXN0ZW06YWRtaW4tdXNlciJ9.MejrO_QQmPOg-
ga5wXatkBBsTD5NbT0GHyIdxK5Ki3L4Yt1ZjTB8cCmhC2cN7kpus6RXN8fZpeB72UohSd1JBOJbJ9QFobSfEXXgKWD9r366hkuqP3lObTUexNDTs
Vlx12WUD6Vp_QAkq8ItIQ3o6xdeA2udhrAB8E55vPhK2PzyuaLHkkT-
87CmG1amdn9mpZGv4FNHUvS7TYHvHs2ShisWZgLsC9hF8t_TngGWcUA5OXqH_5CzdLAYj3f2qXwXmbYiwrHT9T8PL3gchDDDuvhDxjesWqdWRjKY
DU1mJ5oNskEiBQcRF0mOwl5BlZm8VwNAV1CUdKKXeSeI7_cZ6g
ca.crt:     1025 bytes
namespace:  11 bytes
```

Start-up Proxy

> *kubectl proxy*

```
Starting to serve on 127.0.0.1:8001
```

Open your browser

Navigate to:

http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#!/login

Sign in using the token previously retrieved.

blocked URL

blocked URL

Install Sample Pod

> vi nginx-example.yaml

```
    apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2 # tells deployment to run 2 pods matching the template
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: nginx
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
    - port: 80
      nodePort: 31080
      name: nginx
```

> kubectl apply -f nginx-example.yaml

Open your browser to http://localhost:31080/

More Advances Examples

Node Storage

> *mkdir -p ~/k8s/LOCAL_STORAGE*
>
> *vi local-storage.yml*

```
    apiVersion: v1
kind: PersistentVolume
metadata:
  name: local-storage
spec:
  capacity:
    storage: 10Gi
  # volumeMode field requires BlockVolume Alpha feature gate to be enabled.
  volumeMode: Filesystem
  accessModes:
  - ReadWriteOnce
  persistentVolumeReclaimPolicy: Delete
  storageClassName: local-storage
  local:
    path: /Users/<USER>/k8s/LOCAL_STORAGE
  nodeAffinity:
    required:
      nodeSelectorTerms:
      - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
          - docker-for-desktop
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: local-storage-claim
spec:
  storageClassName: local-storage
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
```

Minecraft Server with Resource Declarations

This example defines a minecraft server that uses:

- local storage
- resource management

> *vi minecraft.yml*

```
    apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: minecraft
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: minecraft
    spec:
      containers:
      - name: minecraft
        image: itzg/minecraft-server
        env:
        - name: EULA
          value: "TRUE"
        - name: VERSION
          value: "LATEST"
        - name: OPS
          value: johnmehan
        resources:
          requests:
            memory: "1024Mi"
            cpu: "1000m"
          limits:
            memory: "2048Mi"
            cpu: "2000m"
        ports:
        - containerPort: 25565
        volumeMounts:
        - mountPath: /data
          name: local-vol
          subPath: minecraft/data
      volumes:
      - name: local-vol
        persistentVolumeClaim:
          claimName: local-storage-claim
---
apiVersion: v1
kind: Service
metadata:
  name: minecraft
spec:
  type: NodePort
  selector:
    app: minecraft
  ports:
    - port: 25565
      nodePort: 32556
      name: minecraft
```

Rest Cluster/Start Over

If you want to reset your cluster and start from scratch, you can do this through the Docker UI.

blocked URL

References

Reference	URL
Docker - Deploy on Kubernetes	https://docs.docker.com/docker-for-mac/kubernetes/

Kubernetes Dashboard	https://github.com/kubernetes/dashboard
Getting Started with Kubernetes with Docker on Mac	https://rominirani.com/tutorial-getting-started-with-kubernetes-with-docker-on-mac-7f58467203fd