**install hyperhit and minikube**

brew update

brew install hyperkit

brew install minikube

kubectl

minikube

**create minikube cluster**

minikube start --vm-driver=hyperkit

kubectl get nodes

minikube status

kubectl version

**delete cluster and restart in debug mode**

minikube delete

minikube start --vm-driver=hyperkit --v=7 --alsologtostderr

minikube status

**kubectl commands**

kubectl get nodes

kubectl get pod

kubectl get services

kubectl create deployment nginx-depl --image=nginx

kubectl get deployment

kubectl get replicaset

kubectl edit deployment nginx-depl

**debugging**

kubectl logs {pod-name}

kubectl exec -it {pod-name} -- bin/bash

**create mongo deployment**

kubectl create deployment mongo-depl --image=mongo

kubectl logs mongo-depl-{pod-name}

kubectl describe pod mongo-depl-{pod-name}

**delete deplyoment**

kubectl delete deployment mongo-depl

kubectl delete deployment nginx-depl

**create or edit config file**

vim nginx-deployment.yaml

kubectl apply -f nginx-deployment.yaml

kubectl get pod

kubectl get deployment

**delete with config**

kubectl delete -f nginx-deployment.yaml

#Metrics

kubectl top The kubectl top command returns current CPU and memory usage for a cluster’s pods or nodes, or for a particular pod or node if specified.

**nginx-deployment-result.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

annotations:

deployment.kubernetes.io/revision: "1"

kubectl.kubernetes.io/last-applied-configuration: |

{"apiVersion":"apps/v1","kind":"Deployment","metadata":{"annotations":{},"labels":{"app":"nginx"},"name":"nginx-deployment","namespace":"default"},"spec":{"replicas":2,"selector":{"matchLabels":{"app":"nginx"}},"template":{"metadata":{"labels":{"app":"nginx"}},"spec":{"containers":[{"image":"nginx:1.16","name":"nginx","ports":[{"containerPort":8080}]}]}}}}

creationTimestamp: "2020-01-24T10:54:56Z"

generation: 1

labels:

app: nginx

name: nginx-deployment

namespace: default

resourceVersion: "96574"

selfLink: /apis/apps/v1/namespaces/default/deployments/nginx-deployment

uid: e1075fa3-6468-43d0-83c0-63fede0dae51

spec:

progressDeadlineSeconds: 600

replicas: 2

revisionHistoryLimit: 10

selector:

matchLabels:

app: nginx

strategy:

rollingUpdate:

maxSurge: 25%

maxUnavailable: 25%

type: RollingUpdate

template:

metadata:

creationTimestamp: null

labels:

app: nginx

spec:

containers:

- image: nginx:1.16

imagePullPolicy: IfNotPresent

name: nginx

ports:

- containerPort: 8080

protocol: TCP

resources: {}

terminationMessagePath: /dev/termination-log

terminationMessagePolicy: File

dnsPolicy: ClusterFirst

restartPolicy: Always

schedulerName: default-scheduler

securityContext: {}

terminationGracePeriodSeconds: 30

status:

availableReplicas: 2

conditions:

- lastTransitionTime: "2020-01-24T10:54:59Z"

lastUpdateTime: "2020-01-24T10:54:59Z"

message: Deployment has minimum availability.

reason: MinimumReplicasAvailable

status: "True"

type: Available

- lastTransitionTime: "2020-01-24T10:54:56Z"

lastUpdateTime: "2020-01-24T10:54:59Z"

message: ReplicaSet "nginx-deployment-7d64f4b574" has successfully progressed.

reason: NewReplicaSetAvailable

status: "True"

type: Progressing

observedGeneration: 1

readyReplicas: 2

replicas: 2

updatedReplicas: 2

**nginx-deployment.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

labels:

app: nginx

spec:

replicas: 2

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:1.16

ports:

- containerPort: 8080

**nginx-service.yaml**

apiVersion: v1

kind: Service

metadata:

name: nginx-service

spec:

selector:

app: nginx

ports:

- protocol: TCP

port: 80

targetPort: 8080