**CKA考题**

命令补全：source <(kubectl completion bash)

**第一题------------------------------**



kubectl config use-context k8s

kubectl create **clusterrole** deployment-clusterrole --verb=create --resource=deployments,statefulsets,daemonsets

kubectl create **serviceaccounts** cicd-token -n app-team1

kubectl -n app-team1 create **rolebinding** deployment-rolebinding --clusterrole deployment-clusterrole --serviceaccount app-team1:cicd-token

**第二题------------------------------**



kubectl config use-context ek8s

kubectl **cordon** ek8s-node1

kubectl **drain** ek8s-node1 --delete-local-data=true --ignoredaemonsets=true --force

**第三题------------------------------**



#配置环境

kubectl config use-context mk8s

#开始操作

kubectl get nodes

ssh k8s-master1

sudo -i

kubectl drain k8s-master1 --ignore-daemonsets=true

apt-cache show kubeadm |grep 1.20.1

apt install kubeadm=1.20.1-00 kubelet=1.20.1-00 kubectl=1.20.1-00

kubeadm upgrade apply 1.20.1 --etcd-upgrade=false

kubectl uncordon k8s-master1

kubectl get nodes

#回到student视图下

exit

exit

**第四题------------------------------**



docker ps -a |grep etcd

docker cp e99ee398d40c:/usr/local/bin/etcdctl /usr/bin/

export ETCDCTL\_API=3

#备份

etcdctl --endpoints=<https://127.0.0.1:2379> --cacert="/opt/KUIN00601/ca.cert" --cert="/opt/KUIN00601/etcd-client.crt" --key="/opt/KUIN00601/etcd-client.key" snapshot save /var/lib/backup/etcd-snapshot.db

#还原

etcdctl snapshot restore /var/lib/bakup/etcd-snapshot-previous.db

**第五题------------------------------**



k8s.io官网搜索NetworkPolicy复制第一个yaml

复制第一个yaml

vi NetworkPolicy.yaml

apiVersion: networking.k8s.io/v1

kind: NetworkPolicy

metadata:

name: allow-port-from-namespace

namespace: internal

spec:

podSelector: {}

policyTypes:

- Ingress

ingress:

- from:

  - namespaceSelector:

      matchLabels:

        echo-key: echo-value

  - podSelector: {}

  ports:

  - protocol: TCP

    port: 9000

kubectl apply -f NetworkPolicy.yaml

**第六题------------------------------**



kubectl edit deployment front-end

​

...

spec:

  containers:

  - image: nginx:alpine

    imagePullPolicy: IfNotPresent

    name: nginx

    ports:

    - containerPort: 80

      name: http

      protocol: TCP

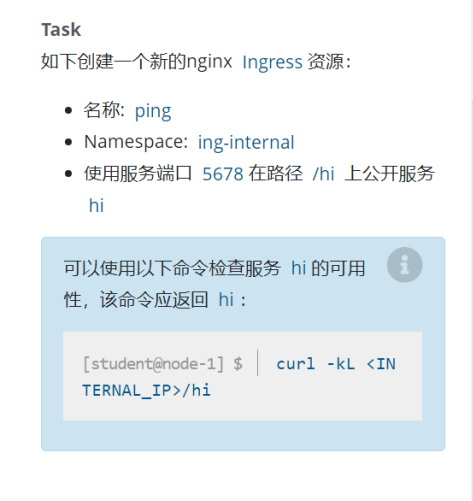
    resources: {}

...

kubectl expose deployment front-end --port=80 --target-port=80 protocol=TCP --name=front-end-svc --type=NodePort

curl -I 127.0.0.1:30971

**第七题------------------------------**



kubectl config use-context k8s

在k8s.io官网搜索ingress复制第一个

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: ping

namespace: ing-internal

spec:

rules:

- http:

    paths:

    - path: /hi

      pathType: Prefix

      backend:

        service:

          name: hi

          port:

            number: 5678

kubectl apply -f ingress.yaml

curl -kL <INTERNAL\_IP>/hi

**第八题------------------------------**



kubectl config use-context k8s

kubectl **scale** --replicas=4 deployment web-server

kubectl get deployment

**第九题------------------------------**



kubectl config use-context k8s

kubectl get nodes --show-labels

# 有可能node已经打好了标签就不需要打了

kubectl label nodes k8s-node1 disk=ssd

# k8s.io官网搜索nodeselector复制第二个yaml

vi pod.yaml

apiVersion: v1

kind: Pod

metadata:

name: nginx-kusc00401

spec:

containers:

- name: nginx

  image: nginx

nodeSelector:

  disk: ssd

kubectl apply -f pod.yaml

**第十题------------------------------**



kubectl config use-context k8s

kubectl get node

kubectl describe node | grep Taint

mkdir -p /opt/KUSC00402

echo 2 > /opt/KUSC00402/kusc00402.txt

**第十一题------------------------------**



kubectl config use-context k8s

#官网搜索nodeselector复制第一个

vim pod2.yaml

apiVersion: v1

kind: Pod

metadata:

name: kucc8

spec:

containers:

- name: nginx

  image: nginx

- name: redis

  image: redis

- name: memcached

  image: memcached

- name: consul

  image: consul

kubectl apply -f pod2.yaml

**第十二题------------------------------**



#k8s.io官网搜索pv再查找关键字hostPath点击进入an example of hostPath typed volume样例页面复制第一个yaml

vim pv.yaml

apiVersion: v1

kind: PersistentVolume

metadata:

name: app-config

spec:

capacity:

  storage: 2Gi

accessModes:

  - ReadOnlyMany

hostPath:

  path: /src/app-config

kubectl apply -f pv.yaml

**第十三题------------------------------**



#官网搜索pv复制PersistentVolumeClaims样例下的改改

vim pvc.yaml

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: pv-volume

spec:

accessModes:

  - ReadWriteOnce

resources:

  requests:

    storage: 10Mi

storageClassName: csi-hostpath-sc

kubectl apply -f pvc.yaml

#回到官网查找Claims As Volumes样例

vim pod-pvc.yaml

apiVersion: v1

kind: Pod

metadata:

name: web-server

spec:

containers:

- name: nginx

  image: nginx

  volumeMounts:

    - mountPath: "/usr/share/nginx/html"

      name: mypd

volumes:

  - name: mypd

    persistentVolumeClaim:

      claimName: pv-volume

kubectl apply -f pod-pvc.yaml

kubectl edit pvc pv-volume

**第十四题------------------------------**



kubectl logs foo | grep file-not-found >> /opt/KUTR00101/foo

**第十五题------------------------------**



k8s.io官网搜索sidecar复制第二个yaml

kubectl get pod 11-factor-app -o yaml > 11-factor-app.yaml

vim 11-factor-app.yaml

kubectl delete pod 11-factor-app

kubectl apply -f 11-factor-app.yaml

kubectl exec 11-factor-app -c sidecar -- tail -f /var/log/11-factor-app.log

**第十六题------------------------------**



kubectl top pod -A -l name=cpu-loader --sort-by='cpu'

echo <pod名称> >> /opt/KUTR00401/KUTR00401.txt

**第十七题------------------------------**



systemctl start kubelet
systemctl enable kubelet