CKA 练习题

Question: 1

监控 Pod foo 的日志，提取出带有 error unable-to-access-website 的行，并写入/opt/xxx/xxx

Solution: 1

kubectl logs foo |grep unable-to-access-website > /opt/xxx/xxx

Question: 2

列出所有PV，使用kubectl 自带的排序方法按capacity 排序

Solution: 2

kubectl get pv --sort-by=spec.capacity.storage > /opt/xxx/xxx

Question: 3

让 pod nginx 运行在集群的每一台主机上，image 名为nginx，不要改动现有的taint。使用 DaemonSets，名称为 xxx。

Solution: 3

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: xxx
  labels:
    k8s-app: ds-xxx
spec:
  selector:
    matchLabels:
      name: xxx
  template:
    metadata:
      labels:
        name: xxx
    spec:
      containers:
```

```
    - name: xxx
      image: nginx
```

Question: 4

- 向已经存在的/opt/xxx/xxx.yaml 中加入init container
- init container 需要创建空文件 /workdir/faithful.txt
- 如果/workdir/faithful.txt 没有被检测到，pod 会退出
- 一旦文件被检测到，pod 将被创建

The /opt/xxx/xxx.yaml:

```
apiVersion: v1
kind: Pod
metadata:
  name: hungry-bear
spec:
  volumes:
    - name: workdir
      emptyDir: {}
  containers:
  - name: checker
    image: alpine
    command: ["/bin/sh", "-c", "if [ -f /workdir/faithful.txt]; then sleep 100000; else exit 1; fi"]
    volumeMounts:
    - name: workdir
      mountPath: /workdir
```

Solution: 4

```
apiVersion: v1
kind: Pod
metadata:
  name: hungry-bear
spec:
  volumes:
    - name: workdir
      emptyDir: {}
  containers:
  - name: checker
    image: alpine
    command: ["/bin/sh", "-c", "if [ -f /workdir/faithful.txt]; then sleep 100000; else exit 1; fi"]
    volumeMounts:
    - name: workdir
      mountPath: /workdir
  initContainers:
```

```
  - name: init-myservice
    image: busybox
    command: ['sh', '-c', 'touch /workdir/faithful.txt']
    volumeMounts:
    - name: workdir
      mountPath: /workdir
```

Question: 5

创建名为xxx 的 pod，包括以下image 的 container：nginx+redis

Solution: 5

```
apiVersion: v1
kind: Pod
metadata:
  name: xxx
spec:
  containers:
    - name: nginx
      image: nginx
    - name: redis
      image: redis
```

Question: 6

创建一个Pod：

- Name: nginx-xxx
- Image: nginx
- Node selector: disk=spinning

Solution: 6

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-xxx
spec:
  containers:
  - name: nginx
    image: nginx
  nodeSelector:
    disk: spinning
```

Question: 7

创建一个deployment:

- Name: nginx-app
- image: nginx, version: 1.11.10-alpine
- 3 副本

用 rolling-update 升级为 1.11.13-alpine 版本，并记录过程。然后回滚至之前的版本 1.11.10-alpine。

Solution: 7

kubectl run nginx-app --image=nginx:1.11.10-alpine --replicas=3
kubectl set image deployment nginx-app nginx-app=nginx:1.11.13-alpine --record
kubectl rollout undo deployment nginx-app

Question: 8

创建 service front-end-service，拥有NodePort，并路由至已存在 pod front-end.

Solution: 8

kubectl expose deployment front-end --type=NodePort --name=front-end-service

Question: 9

创建 pod：

- Name: nginx
- Using image: nginx
- In a new namespace: website-backend.

Solution: 9

kubectl create ns website-backend

```
---
apiVersion: v1
kind: Pod
metadata:
   name: nginx
   namespace: website-backend
spec:
   containers:
   - name: nginx
      image: nginx
```

Question: 10

创建一个deployment:

- 3 副本的redis 镜像，label: pipeline_stage=test
- name: xxx

保存配置文件至/opt/xxx.yaml (or .json).
完成后清理本题的所有资源对象

Solution: 10

kubectl run xxx --image=redis --replicas=3 -l pipeline_stage=test
kubectl get deployment xxx -o yaml > /opt/xxx.yaml
kubectl delete deployment xxx

Question: 11

列出所有与Service bar 相连的pods

Solution: 11

kubectl get svc bar -o yaml
get the key-value in the "selector", e.g. app=bar
kubectl get pod -l app=bar |awk '{print $1}' > /opt/xxx.txt

Question: 12

创建一个 Secret:

- Name: super-secret
- credential: bob

创建一个名为pod-secrets-via-file 的 pod，使用redis image, 挂载super-secret 在/secrets
创建一个名为pod-secrets-via-env 的 pod，使用redis image, CREDENTIALS 作为环境变量

Solution: 12

$ echo -n 'bob' | base64
Ym9i
---
apiVersion: v1
kind: Secret
metadata:
  name: super-secret
type: Opaque
data:
  credential: Ym9i

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-secrets-via-file
spec:
  containers:
  - name: mypod
    image: redis
    volumeMounts:
    - name: foo
      mountPath: "/secrets"
  volumes:
  - name: foo
    secret:
      secretName: super-secret
---
apiVersion: v1
kind: Pod
metadata:
  name: pod-secrets-via-env
spec:
  containers:
  - name: mycontainer
    image: redis
    env:
      - name: CREDENTIALS
        valueFrom:
          secretKeyRef:
            name: super-secret
            key: credential
```

Question: 13

创建 pod：

image: redis

volume 名：app-cache

挂载位置：/data/redis

在 qa namespace 中，卷不能是持久化的。

Solution: 13

```
apiVersion: v1
kind: Pod
metadata:
```

```
    name: non-persistent-redis
    namespace: qa
spec:
    containers:
    - image: redis
        name: test-container
        volumeMounts:
        - mountPath: /data/redis
            name: app-cache
    volumes:
    - name: app-cache
        emptyDir: {}
```

Question: 14

把 deployment guestbook 变为 3 副本

Solution: 14

kubectl scale deployment guestbook --replicas=3

Question: 15

查看有几个node 是 ready 的，不包括tainted 节点。

Solution: 15

kubectl get nodes
kubectl describe nodes |grep -i taint
get the number of nodes that are Ready and not NoSchedule

Question: 16

从标签为name=cpu-loader 的 pod 中，找到最高CPU 使用的pod

Solution: 16

kubectl top pod -l name=cpu-loader

Question: 17

创建一个deployment:

- Name: nginx-dns
- 连接的service: nginx-dns

- 确保 service 和 pod 可以从他们的DNS record 进入
- 使用 nginx 镜像

使用 nslookup 查出service 和 pod 的 DNS record

Solution: 17

kubectl run nginx-dns --image=nginx
kubectl expose deployment nginx-dns --port=80
kubectl get pod -o wide
get the IP address of nginx-dns pod, e.g. 1.2.3.4
kubectl run curl --image=busybox:1.28 -it

nslookup nginx-dns
nslookup 1-2-3-4.default.pod.cluster.local

Question: 18

创建etcd 快照存到/var/lib/backup/etcd-snapshot.db，版本为 3.1.10，

- https://127.0.0.1:2379
- CA certificate: /opt/xxx/ca.crt
- Client certificate: /opt/xxx/etcd-client.crt
- Client key: /opt/xxx/etcd-client.key

Solution: 18

```
ETCDCTL_API=3 etcdctl \
  --endpoints=https://127.0.0.1:2379 \
  --cacert=/opt/xxx/ca.crt \
  --cert=/opt/xxx/etcd-client.crt \
  --key=/opt/xxx/etcd-client.key \
  snapshot save /var/lib/backup/etcd-snapshot.db
```

Question: 19

设定 node0 不可用并reschedule 其上的所有pod。

Solution: 19

kubectl config use-context ek8s
kubectl taint nodes ek8s-node-0 key=value:NoSchedule
kubectl get pod -o wide |grep ek8s-node-0
get the pods on ek8s-node-0 and delete them
kubectl delete pod xxx

Question: 20

工作节点node0 状态为NotReady，找出原因并永久性修复。

Solution: 20

kubectl config use-context wk8s
ssh wk8s-node-0
sudo -i
systemctl restart kubelet
systemctl enable kubelet

Question: 21

配置 node1 的 kubelet 的 systemd，自动启动一个Jenkins 镜像的pod，配置文件放入节点的/etc/kubernetes/manifests。

Solution: 21

kubectl config use-context wk8s
ssh wk8s-node-1
sudo -i
systemctl status kubelet
get the systemd spec file, e.g. /etc/systemd/system/kubelet.service, and add a line "--pod-manifest-path=/etc/kubernetes/manifests" to ExecStart
vim /etc/systemd/system/kubelet.service
--pod-manifest-path=/etc/kubernetes/manifests
---
vim /etc/kubernetes/manifests/jenkins.yaml
apiVersion: v1
kind: Pod
metadata:
  name: myservice
spec:
  containers:
  - name: myservice
    image: jenkins
---
systemctl restart kubelet

Question: 22

给 Kubernetes 集群加节点，安装并启动 kubelet。（过于复杂建议略过）

Solution: 22

Reference: https://kubernetes.io/docs/reference/command-line-tools-reference/kubelet-tls-bootstrapping/

Question: 23

修复失效的service，永久性恢复集群健康。

Solution: 23

kubectl get cs

get the Unhealthy component

ssh bk8s-master-0

systemctl restart kube-controller-manager

systemctl enable kube-controller-manager

Question: 24

创建持久化卷，容量为 1Gi，模式为ReadWriteMany，类型为hostPath 的位置在/xxx/xxx。

Solution: 24

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: xxx
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: /xxx/xxx
```