

CKA考试题目与解答

Original桶装酱油王 桶装酱油王 今天

重点来了。由于考试不允许将考题记录在考试系统外，凭着自己的记忆，加上浏览器历史记录，记录下21题（总共24题），内容有部分差异，考题中的pod之类的镜像、命名肯定都记不清了，凭空造的，请见谅。

题目1

列出环境内所有的pv 并以 name字段排序（使用kubectl自带排序功能）

命令：

```
kubectl get pv --sort-by=.metadata.name
```

考点：kubectl命令熟悉程度

参考：kubectl cheatsheet

链接：<https://kubernetes.io/docs/reference/kubectl/cheatsheet/>

题目2

列出指定pod的日志中状态为Error的行，并记录在指定的文件上

命令：

```
kubectl logs <podname> | grep bash > /opt/KUCC000xxx/KUCC000xxx.txt
```

考点：Monitor, Log, and Debug

参考：Debugging Pods

链接：<https://kubernetes.io/docs/tasks/debug-application-cluster/debug-pod-replication->

题目3

列出k8s可用的节点，不包含不可调度的 和 NoReachable的节点，并把数字写入到文件里

命令：

```
#笨方法，人工数
kubectl get nodes

#CheatSheet方法，应该还能优化JSONPATH
JSONPATH='{range .items[*]}{@.metadata.name}:{range @.status.conditions[*]}{@.type}={@.status};{end}{end}' \
&& kubectl get nodes -o jsonpath="$JSONPATH" | grep "Ready=True"
```

考点：kubectl命令熟悉程度

参考：kubectl cheatsheet

链接：<https://kubernetes.io/docs/reference/kubectl/cheatsheet/>

题目4

创建一个pod名称为nginx，并将其调度到节点为 disk=stat上

命令：

```
#这是我的操作,实际上从文档复制更快
kubectl run nginx --image=nginx --restart=Never --dry-run > 4.yaml
#增加对应参数
vi 4.yaml
kubectl apply -f 4.yaml
```

YAML：

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

```
name: nginx
labels:
  env: test
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  nodeSelector:
    disktype: sata
```

考点：pod的调度。

参考：assign-pod-node

链接：<https://kubernetes.io/docs/concepts/configuration/assign-pod-node/>

题目5

提供一个pod的yaml，要求添加Init Container，Init Container的作用是创建一个空文件，pod的Containers判断文件是否存在，不存在则退出

YAML:

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: apline
    image: nginx
    com-
mand: ['sh', '-c', 'if [目录下有work文件];then sleep 3600; else exit; fi;']
###增加init Container####
initContainers:
- name: init
  image: busybox
  command: ['sh', '-c', 'touch /目录/work;']
```

考点：init Container。一开始审题不仔细，以为要用到livenessProbes

参考：init-containers

链接：<https://kubernetes.io/docs/concepts/workloads/pods/init-containers/>

题目6

指定在命名空间内创建一个pod名称为test，内含四个指定的镜像nginx、redis、memcached、busybox

命令：

```
kubectl run test --image=nginx --image=redis --image=memcached --image=busybox --restart=Never -n <namespace>
```

考点：kubectl命令熟悉程度、多个容器的pod的创建

参考：kubectl cheatsheet

链接：<https://kubernetes.io/docs/reference/kubectl/cheatsheet/>

题目7

创建一个pod名称为test，镜像为nginx，Volume名称cache-volume为挂在在/data目录下，且Volume是non-Persistent的

YAML：

```
apiVersion: v1
kind: Pod
metadata:
  name: test
spec:
  containers:
    - image: nginx
      name: test-container
      volumeMounts:
        - mountPath: /cache
          name: cache-volume
  volumes:
    - name: cache-volume
      emptyDir: {}
```

考点：Volume、emptydir

参考: Volumes

链接: <https://kubernetes.io/docs/concepts/storage/volumes/#emptydir>

题目8

列出Service名为test下的pod 并找出使用CPU使用率最高的一个, 将pod名称写入文件中

命令:

```
#使用-o wide 获取service test的SELECTOR
kubectl get svc test -o wide
##获取结果我就随便造了
NAME                                TYPE                CLUSTER-IP      EXTER-
NAL-IP      PORT(S)          AGE          SELECTOR
test        ClusterIP        None           <none>        3306/TCP    50d          app=word-
press,tier=mysql

#获取对应SELECTOR的pod使用率, 找到最大那个写入文件中
kubectl top test -l 'app=wordpress,tier=mysql'
```

考点: 获取service selector, kubectl top监控pod资源

参考: Tools for Monitoring Resources

链接: <https://kubernetes.io/docs/tasks/debug-application-cluster/resource-usage-monitoring>

题目9

创建一个Pod名称为nginx-app, 镜像为nginx, 并根据pod创建名为nginx-app的Service, type为NodePort

命令:

```
kubectl run nginx-app --image=nginx --restart=Never --port=80
kubectl create svc nodeport nginx-app --tcp=80:80 --dry-run -o yaml > 9.yaml
#修改yaml, 保证selector name=nginx-app
vi 9.yaml
```

YAML:

```
apiVersion: v1
kind: Service
metadata:
  creationTimestamp: null
  labels:
    app: nginx-app
  name: nginx-app
spec:
  ports:
    - name: 80-80
      port: 80
      protocol: TCP
      targetPort: 80
  selector:
    #注意要和pod对应
    name: nginx-app
  type: NodePort
```

考点: Service

参考: publishing-services-service-types

链接: <https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types>

题目10

创建一个nginx的Workload，保证其在每个节点上运行，注意不要覆盖节点原有的Tolerations

这道题直接复制文档的yaml太长了，由于daemonSet的格式和Deployment格式差不多，我用旁门左道的方法 先创建Deploy，再修改成daemonset，这样速度会快一点

```
#先创建一个deployment
kubectl run nginx --image=nginx --dry-run -o yaml > 10.yaml
#将yaml改成DaemonSet
vi 10.yaml
```

YAML:

```
#修改apiVersion和kind
#apiVersion: extensions/v1beta1
#kind: Deployment
```

```
apiVersion:apps/v1
kind: DaemonSet
metadata:
  creationTimestamp: null
  labels:
    run: nginx
  name: nginx
spec:
#去掉replicas
# replicas: 1
  selector:
    matchLabels:
      run: nginx
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        run: nginx
    spec:
      containers:
      - image: nginx
        name: nginx
        resources: {}
status: {}
```

考点：DaemonSet

参考：DaemonSet

链接：<https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/>

题目11

将deployment为nginx-app的副本数从1变成4。

命令：

```
#方法1
kubectl scale --replicas=4 deployment nginx-app
#方法2，使用edit命令将replicas改成4
kubectl edit deploy nginx-app
```

考点：deployment的Scaling，搜索Scaling

参考：Scaling the application by increasing the replica count

链接: <https://kubernetes.io/docs/tasks/run-application/run-stateless-application-deployment/#scaling-the-application-by-increasing-the-replica-count>

题目12

创建nginx-app的deployment, 使用镜像为nginx:1.11.0-alpine, 修改镜像为1.11.3-alpine, 并记录升级, 再使用回滚, 将镜像回滚至nginx:1.11.0-alpine

命令:

```
kubectl run nginx-app --image=nginx:1.11.0-alpine
kubectl set image deployment nginx-app --image=nginx:1.11.3-alpine
kubectl rollout undo deployment nginx-app
kubectl rollout status -w deployment nginx-app
```

考点: 资源的更新

参考: Kubectl Cheat Sheet: Updating Resources

链接: <https://kubernetes.io/docs/reference/kubectl/cheatsheet/#updating-resources>

题目13

根据已有的一个nginx的pod、创建名为nginx的svc、并使用nslookup查找出service dns记录, pod的dns记录并分别写入到指定的文件中

命令:

```
#创建一个服务
kubectl create svc nodeport nginx --tcp=80:80
#创建一个指定版本的busybox, 用于执行nslookup
kubectl create -f https://k8s.io/examples/admin/dns/busybox.yaml
#将svc的dns记录写入文件中
kubectl exec -ti busybox -- nslookup nginx > 指定文件
#获取pod的ip地址
kubectl get pod nginx -o yaml
#将获取的pod ip地址使用nslookup查找dns记录
kubectl exec -ti busybox -- nslookup <Pod ip>
```

考点: 网络相关, DNS解析

参考: Debugging DNS Resolution

链接: <https://kubernetes.io/docs/tasks/administer-cluster/dns-debugging-resolution>

题目14

创建Secret 名为mysecret, 内含有password字段, 值为bob, 然后 在pod1里 使用ENV进行调用, Pod2里使用Volume挂载在/data 下

命令:

```
#将密码值使用base64加密,记录在Notepad里  
echo -n 'bob' | base64
```

YAML:

14.secret.yaml

```
apiVersion: v1  
kind: Secret  
metadata:  
  name: mysecret  
type: Opaque  
data:  
  password: Ym9i
```

14.pod1.yaml

```
apiVersion: v1  
kind: Pod  
metadata:  
  name: pod1  
spec:  
  containers:  
    - name: mypod  
      image: nginx  
      volumeMounts:  
        - name: mysecret  
          mountPath: "/data"  
          readOnly: true  
  volumes:  
    - name: mysecret  
      secret:
```

```
secretName: mysecret
```

14.pod2.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: pod2
spec:
  containers:
  - name: mycontainer
    image: redis
    env:
    - name: SECRET_PASSWORD
      valueFrom:
        secretKeyRef:
          name: mysecret
          key: password
```

考点 Secret

参考: Secret

链接: <https://kubernetes.io/docs/concepts/configuration/secret/>

题目15

使node1节点不可调度，并重新分配该节点上的pod

命令:

```
#直接drain会出错，需要添加--ignore-daemonsets --delete-local-data参数
kubectl drain node node1 --ignore-daemonsets --delete-local-data
```

考点: 节点调度、维护

参考: [Safely Drain a Node while Respecting Application SLOs]: (<https://kubernetes.io/docs/tasks/administer-cluster/safely-drain-node/>)

链接: <https://kubernetes.io/docs/tasks/administer-cluster/safely-drain-node/>

题目16

使用etcd 备份功能备份etcd（提供endpoints, ca、cert、key）

命令：

```
ETCDCTL_API=3 etcdctl --endpoints https://127.0.0.1:2379 \
--cacert=ca.pem --cert=cert.pem --key=key.pem \
snapshot save snapshotdb
```

考点：etcd的集群的备份与恢复

参考：backing up an etcd cluster

链接：<https://kubernetes.io/docs/tasks/administer-cluster/configure-upgrade-etcd/#backing-up-an-etcd-cluster>

题目17

给出一个失联节点的集群，排查节点故障，要保证改动是永久的。

命令：

```
#查看集群状态
kubectl get nodes
#查看故障节点信息
kubectl describe node node1

#Message显示kubelet无法访问（记不清了）
#进入故障节点
ssh node1

#查看节点中的kubelet进程
ps -aux | grep kubelete
#没找到kubelet进程，查看kubelet服务状态
systemctl status kubelet.service
#kubelet服务没启动，启动服务并观察
systemctl start kubelet.service
#启动正常，enable服务
systemctl enable kubelet.service

#回到考试节点并查看状态
exit

kubectl get nodes #正常
```

考点：故障排查

参考：Troubleshoot Clusters

链接：<https://kubernetes.io/docs/tasks/debug-application-cluster/debug-cluster/>

题目18

给出一个集群，排查出集群的故障

这道题没空做完。kubectl get node显示connection refuse，估计是apiserver的故障。

考点：故障排查

参考：Troubleshoot Clusters

链接：<https://kubernetes.io/docs/tasks/debug-application-cluster/debug-cluster/>

题目19

给出一个节点，完善kubelet配置文件，要求使用systemd配置kubelet

这道题没空做完

考点我知道，doc没找到..在哪 逃~ε=ε=ε= ∟(° □ ° ;)_

题目20

给出一个集群，将节点node1添加到集群中，并使用TLS bootstrapping

这道题没空做完，花费时间比较长，可惜了。

考点：TLS Bootstrapping

参考：TLS Bootstrapping

链接：<https://kubernetes.io/docs/reference/command-line-tools-reference/kubelet-tls-bootstrapping/>

创建一个pv，类型是hostPath，位于/data中，大小1G，模式ReadOnlyMany

YAML:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadOnlyMany
  hostPath:
    path: /data
```

考点：创建PV

参考：persistent volumes

链接：<https://kubernetes.io/docs/concepts/storage/persistent-volumes/>

最后再次感谢一下**华为7天kubernetes管理员实训课程**、对于完全没有接触过k8s的同学，难度是高了。但是作为有k8s基础，备考CKA的同学是非常非常有用的。基本上讲的考点都有了。