CKA考题解析

第1题

创建一个名为deployment-clusterrole且仅允许创建以下资源类型的新ClusterRole:

* Deployment
* StatefulSet
* DaemonSet

在现有的namespace app-team1中创建一个名为cicd-token的新ServiceAccount。

限于namespace app-team1，将新的ClusterRole deployment-clusterrole绑定到新的ServiceAccount cicd-token

解题：

|  |
| --- |
| vim cka-01-clusterrole.yaml  apiVersion: rbac.authorization.k8s.io/v1  kind: ClusterRole  metadata:  name: deployment-clusterrole  rules:  - apiGroups:  - apps  resources:  - daemonsets  - deployments  - statefulsets  verbs:  - create  kubectl apply -f cka-01-clusterrole.yaml  kubectl create ns app-team1  kubectl create sa cicd-token -n app-team1  kubectl create rolebinding cicd-token-rolebinding --serviceaccount=app-tema1:cicd-token --clusterrole=deployment-clusterrole -n app-team1  kubectl delete deployment test-busybox -n app-team1 |

第2题

将一个名为ek8s-node-1的节点设置为不可用并将其上的pod重新调度

解题：

|  |
| --- |
| kubectl drain ek8s-node-1 --ignore-daemonsets |

第3题

Given an existing kubernetes cluster running version 1.20.5, upgrade all of the kubernetes control plain and node components on the master node only to version 1.21.0.

you are also expected to upgrade kubelet and kubectl on the master node

tips: Be sure to drain the master node before upgrading it and uncordon it after the upgrade.

Do not upgrade the worker nodes,etcd,the container manager,the CNI plugin, the DNS service or any other addons.

解题：

|  |
| --- |
| kubectl drain cka01 --ignore-daemonsets  apt update -y  apt-cache madison kubeadm  apt upgrade kubeadm=1.21.0-00 kubelet=1.21.0-00 kubectl=1.21.0-00 -y  kubeadm version  kubeadm upgrade plan  kubeadm upgrade apply v1.21.0 --etcd-upgrade=false  ###  ###[upgrade/successful] SUCCESS! Your cluster was upgraded to "v1.20.2". Enjoy!  ###[upgrade/kubelet] Now that your control plane is upgraded, please proceed with upgrading your kubelets if you haven't already done so.  ###  # 回退coredns  coredns:v1.8 --> coredns:1.7.0  kubectl uncordon cka01 |

第4题

首先，为运行在https://127.0.0.1:2379上的现有etcd实例创建快照并将快照保存至/data/bucket/etcd-snapshot.db

然后还原位于/srv/data/etcd-snaphot-previous.db的现有先前快照

提示： 为给定实例创建快照预计在几秒内完成。如果该操作似乎挂起，则命令可能有问题。用ctrl+c来取消操作，然后重试。

提供了以下TLS证书和密钥，以通过etcdctl连接到服务器

* CA证书：/opt/KUIN00601/ca.crt
* 客户端证书： /opt/KUIN00601/etcd-client.crt
* 客户端密钥：/opt/KUIN00601/etcd-client.key

解题：

|  |
| --- |
| export ETCDCTL\_API=3  etcdctl --endpoints=https://127.0.0.1:2379 --cacert=/opt/KUIN00601/ca.crt --cert=/opt/KUIN00601/etcd-client.crt --key=/opt/KUIN00601/etcd-client.key snapshot save /data/backup/etcd-snapshot.db  # etcdctl snapshot restore /data/backup/etcd-snapshot.db  # rm -rf default.etcd  etcdctl snapshot restore /srv/data/etcd-snapshot-previous.db  sudo systemctl stop etcd  # owner: etcd  ll /var/lib/etcd -d  mv /var/lib/etcd /tmp/etcd.bak  mv ~/default.etcd /var/lib/etcd  chown etcd.etcd -R /var/lib/etcd  sudo systemctl start etcd |

第5题

创建一个名为allow-port-from-namespace的新NetworkPolicy，以允许现有namespace internal中的Pods连接到同一namespace中其他Pods的端口8080。

确保新的NetworkPolicy：

* 不允许对没有在监听端口8080的pods的访问
* 不允许不来自namespace internal的pods的访问

解题：

|  |
| --- |
| apiVersion: networking.k8s.io/v1  kind: NetworkPolicy  metadata:  name: allow-port-from-namespace  namespace: internal  spec:  podSelector: {}  policyTypes:  - Ingress  - Egress  ingress:  - from:  - podSelector: {}  ports:  - protocol: TCP  port: 8080  - protocol: UDP  port: 8080  egress:  - to:  - podSelector: {}  ports:  - protocol: TCP  port: 8080  - protocol: UDP  port: 8080  kubectl create ns internal  kubectl apply -f allow-port-from-namespace.yaml |

第5题2



第6题

Reconfigure the existing deployment front-end and add a port specification named http exposing port 80/tcp of existing container nginx.

Create a new service named front-end-svc exposing the container port http.

Configure the new service to also expose the individual Pods via a NodePort on the nodes on which they are scheduled

解题：

|  |
| --- |
| #front-end.yaml  apiVersion: apps/v1  kind: Deployment  metadata:  labels:  app: front-end  name: front-end  spec:  replicas: 1  selector:  matchLabels:  app: front-end  strategy: {}  template:  metadata:  labels:  app: front-end  spec:  containers:  - image: nginx  name: nginx  resources: {}  # front-end.svc.yaml  apiVersion: v1  kind: Service  metadata:  labels:  app: front-end-svc  name: front-end-svc  spec:  ports:  - name: 80-80  port: 80  protocol: TCP  targetPort: 80  selector:  app: front-end  type: NodePort  kubectl edit deploy front-end  ```  ports:  - name: http  containerPort: 80  protocol: TCP  ```  kubectl apply -f front-end-svc.yaml |

第7题

Create a new nginx ingress resource as follows:

* Name: ping
* Namespace: ing-internal
* Exposing service hi on path /hi using service port 5678

tips: The availability of service hi can be checked using the following commands,which should retun hi:

curl -KL <INTERNAL\_IP>/hi

解题：

|  |
| --- |
| # config environment  kubectl create ns ing-internal  kubectl run hi --image=registry.cn-zhangjiakou.aliyuncs.com/breezey/ping -n ing-internal  kubectl expose pod hi --port=5678 -n ing-internal  # answer  # ping.yaml  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 ping.yaml |

第8题

Scale the deploy persentation to 3 pods

解题：

|  |
| --- |
| # config env  kubectl create deploy presentation --image=busybox -- sleep 3600  # answer  kubectl scale --replicas=3 deploy/presentation |

第9题

Schedule a pod as follows:

* Name: nginx-kusc00401
* Image: nginx
* Node selector: disk=spinning

解题：

|  |
| --- |
| # config env  kubectl label node cka03 disk=spinning  # answer  kubectl run nginx-kusc00401 --image=nginx --dry-run=client -o yaml > nginx-kusc00401.yaml  # modify yaml  # nginx-kusc00401.yaml  apiVersion: v1  kind: Pod  metadata:  labels:  run: nginx-kusc00401  name: nginx-kusc00401  spec:  nodeSelector:  disk: spinning  containers:  - image: nginx  name: nginx-kusc00401  resources: {}  dnsPolicy: ClusterFirst  restartPolicy: Always  kubectl apply -f nginx-kusc00401.yaml |

第10题

Check to see how many nodes are ready(not including nodes tainted NoSchedule) and write the number to /opt/KUSCoo402/kusc00402.txt.

解题：

|  |
| --- |
| for i in `kubectl get nodes | awk '$2 ~/^Ready/{print $1}'`;do kubectl describe node $i |grep Taints |grep "<none>";done | wc -l |

第11题

Create a pod named kucc8 with a single app container for each of the following images running inside(there may be between 1 and 4 images specified):

nginx + redis + memcached + consul

解题：

|  |
| --- |
| kubectl run kucc8 --image=nginx --dry-run=client -o yaml > kucc8.yaml  # kucc8.yaml  apiVersion: v1  kind: Pod  metadata:  creationTimestamp: null  labels:  run: kucc8  name: kucc8  spec:  containers:  - image: nginx  name: nginx  - image: redis  name: redis  - image: memcached  name: memcached  - image: consul  name: consul  resources: {}  dnsPolicy: ClusterFirst  restartPolicy: Always  status: {}  kubectl apply -f kucc8.yaml |

第12题

Create a persistent volume with name app-config, of capacity 1Gi and access mode ReadOnlyMany. The type of volume is hostPath and its location is /srv/app-config.

解题：

|  |
| --- |
| # app-config.yaml  apiVersion: v1  kind: PersistentVolume  metadata:  name: app-config  spec:  capacity:  storage: 1Gi  accessModes:  - ReadOnlyMany  hostPath:  path: /srv/app-config  kubectl apply -f app-config.yaml |

第13题

Create a new PersistentVolumeClaim:

* Name: pv-volume
* Class: csi-hostpath-sc
* Capacity: 10Mi

Create a new Pod which mounts the PersistentVolumeClaim as a volume:

* Name: web-server
* Image: nginx
* Mount path: /usr/share/nginx/html

Configure the new Pod to have ReadWriteOnce access on the volume.

Finally, using kubectl edit or kubectl patch expand the PersistentVolumeClaim to a capacity of 70Mi and record that change

解题：



|  |
| --- |
| # config env  # deploy nfs  apt install -y nfs-kernel-server  vim /etc/exports  /data \*(rw,no\_root\_squash)  mkdir /data  systemctl restart nfs-server  showmount -e nfs-server  chmod 777 -R /data  # deploy nfs-csi  apt install -y lrzsz  # upload nfs-csi.zip  unzip nfs-csi.zip  cd nfs-csi/deploy  kubectl apply -f ./  cd ../  kubectl apply -f csi-hostpath-sc.yaml  kubectl get sc  # answer  # pv-volume.yaml  apiVersion: v1  kind: PersistentVolumeClaim  metadata:  name: pv-volume  spec:  accessModes:  - ReadWriteOnce  resources:  requests:  storage: 10Mi  storageClassName: csi-hostpath-sc  kubectl apply -f pv-volume.yaml  kubectl run web-server --image=nginx --dry-run=client -o yaml > web-server.yaml  # web-server.yaml  apiVersion: v1  kind: Pod  metadata:  creationTimestamp: null  labels:  run: web-server  name: web-server  spec:  volumes:  - name: pv-volume  persistentVolumeClaim:  claimName: pv-volume  containers:  - image: nginx  name: web-server  volumeMounts:  - mountPath: /usr/share/nginx/html  name: pv-volume  resources: {}  dnsPolicy: ClusterFirst  restartPolicy: Always  status: {}  kubectl apply -f web-server.yaml  # 修改为70Mi  kubectl edit pvc pv-volume --record=true |

第14题

Monitor the logs of pod bar and:

* Extract log lines corresponding to error unable-to-access-website
* Write them to /opt/KUTR00101/bar

解题：

|  |
| --- |
| # config env  kubectl run bar --image=registry.cn-zhangjiakou.aliyuncs.com/breezey/bar  # answer  kubectl logs bar |grep unable-to-access-website > /tmp/bar |

第15题

Add a busybox sidecar container to the existing Pod big-corp-app.The new sidecar container has to run the following command:

/bin/sh -c tail -n+1 /var/log/big-corp-app.log

Use a volume mount named logs to make the file /var/log/big-corp-app.log available to the sidecar container

warn: Don't modify the existing container. Don't modify the path of the log file. both containers must access it at /var/log/big-corp-app.log

解题：

|  |
| --- |
| # config env  kubectl run big-corp-app --image=registry.cn-zhangjiakou.aliyuncs.com/breezey/bar  # answer  # big-corp-app.yaml  apiVersion: v1  kind: Pod  metadata:  creationTimestamp: null  labels:  run: big-corp-app  name: big-corp-app  spec:  volumes:  - name: logs  emptyDir:  containers:  - image: registry.cn-zhangjiakou.aliyuncs.com/breezey/bar  name: big-corp-app  volumeMounts:  - name: logs  mountPath: /var/log  resources: {}  - name: busybox  image: busybox  volumeMounts:  - name: logs  mountPath: /var/log  command:  - "/bin/sh"  - "-c"  - "tail -n+1 /var/log/big-corp-app.log"  dnsPolicy: ClusterFirst  restartPolicy: Always  kubectl apply -f big-corp-app.yaml |

第16题

From the pod label name=cpu-loader, find pods running high CPU workloads and write the name of the pod consuming most CPU to file /opt/KUTR00401.txt(which alreay exists).

解题：

|  |
| --- |
| # config env  kubectl create deploy cpu-loader --image=mysql --replicas=5 --dry-run=client -o yaml > cpu-loader.yaml  apiVersion: apps/v1  kind: Deployment  metadata:  creationTimestamp: null  labels:  name: cpu-loader  name: cpu-loader  spec:  replicas: 5  selector:  matchLabels:  name: cpu-loader  strategy: {}  template:  metadata:  creationTimestamp: null  labels:  name: cpu-loader  spec:  containers:  - image: mysql  name: mysql  env:  - name: MYSQL\_ROOT\_PASSWORD  value: wordpress  resources: {}  status: {}  kubectl apply -f cpu-loader.yaml  # answer  kubectl top pods -l name=cpu-loader | sort -k2 -nr | head -1 | awk '{print $1}' > /tmp/cpu-loader.txt |

第17题

A kubernetes worker node, named wk8s-node-0 is in state NotReady. Investigate why this is the case, and perform any appropriate steps to bring the node to a Ready state,ensuring that any changes are made permanent.

tips:

you can ssh to the failed node using:

ssh wk8s-node-0

you can assume elevated privileges on the node with the following command：

sudo -i

解题：

|  |
| --- |
| systemctl start docker  systemctl start kubelet  systemctl enable kubelet docker |