**CKA Exam Topics & Questions with Solutions**

* **Cluster Setup & Configuration**
* **Workloads & Scheduling**
* **Networking & Services**
* **Storage & Persistent Volumes**
* **Security & RBAC**
* **Troubleshooting & Cluster Maintenance**

**1️. Cluster Installation using kubeadm**

**📝 Question:**  
Install a single-node Kubernetes cluster using kubeadm and set up networking using Calico.

**🔹 Solution:**

**Step 1: Initialize the Cluster**

sudo kubeadm init --pod-network-cidr=192.168.0.0/16

**Step 2: Set up kubectl for your user**

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

**Step 3: Install Calico Networking**

kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml

**Step 4: Verify the Cluster**

kubectl get nodes

**2️. Troubleshooting Pods (CrashLoopBackOff)**

**📝 Question:**  
A pod named nginx-pod is stuck in CrashLoopBackOff. Find and fix the issue.

**🔹 Solution:**

**Step 1: Check pod logs**

kubectl logs nginx-pod

**Step 2: Describe the pod to check events**

kubectl describe pod nginx-pod

**Step 3: Common fixes**

* If the container image is incorrect:

kubectl set image pod/nginx-pod nginx=nginx:latest

* If the pod spec is wrong, edit it:

kubectl edit pod nginx-pod

**3️. Exposing a Deployment using a Service**

**📝 Question:**  
Expose a deployment named web-deploy as a **NodePort** service on port 30080.

**🔹 Solution:**

**Step 1: Expose the deployment**

kubectl expose deployment web-deploy --type=NodePort --port=80 --name=web-service

**Step 2: Get the assigned NodePort**

kubectl get svc web-service -o=jsonpath='{.spec.ports[0].nodePort}'

**Step 3: Access the service**

curl http://<NODE-IP>:<NODEPORT>

**4️. Scheduling Pods on Specific Nodes**

**📝 Question:**  
Schedule a pod on a node named node1 using a **nodeSelector**.

**🔹 Solution:**

**Step 1: Create a pod with a nodeSelector**

apiVersion: v1

kind: Pod

metadata:

name: nginx-pod

spec:

nodeSelector:

kubernetes.io/hostname: node1

containers:

- name: nginx

image: nginx

**Step 2: Apply the pod**

kubectl apply -f pod.yaml

**Step 3: Verify the node assignment**

kubectl get pod nginx-pod -o wide

**5️. Managing Role-Based Access Control (RBAC)**

**📝 Question:**  
Create an RBAC Role that allows a user dev-user to list pods in the dev namespace.

**🔹 Solution:**

**Step 1: Create a Role**

apiVersion: rbac.authorization.k8s.io/v1

kind: Role

metadata:

namespace: dev

name: pod-reader

rules:

- apiGroups: [""]

resources: ["pods"]

verbs: ["get", "list"]

**Step 2: Apply the Role**

kubectl apply -f role.yaml

**Step 3: Bind the Role to a user**

apiVersion: rbac.authorization.k8s.io/v1

kind: RoleBinding

metadata:

name: read-pods

namespace: dev

subjects:

- kind: User

name: dev-user

apiGroup: rbac.authorization.k8s.io

roleRef:

kind: Role

name: pod-reader

apiGroup: rbac.authorization.k8s.io

**Step 4: Apply the RoleBinding**

kubectl apply -f rolebinding.yaml

**6️. Creating & Restoring an ETCD Backup**

**📝 Question:**  
Take a backup of etcd and restore it.

**🔹 Solution:**

**Step 1: Take an etcd snapshot**

ETCDCTL\_API=3 etcdctl snapshot save /tmp/etcd-backup.db \

--endpoints=https://127.0.0.1:2379 \

--cacert=/etc/kubernetes/pki/etcd/ca.crt \

--cert=/etc/kubernetes/pki/etcd/server.crt \

--key=/etc/kubernetes/pki/etcd/server.key

**Step 2: Restore etcd from backup**

sh

CopyEdit

ETCDCTL\_API=3 etcdctl snapshot restore /tmp/etcd-backup.db --data-dir /var/lib/etcd

**7️. Configuring Persistent Storage**

**📝 Question:**  
Create a **PersistentVolumeClaim (PVC)** for 2Gi storage.

**🔹 Solution:**

**Step 1: Create the PVC**

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: my-pvc

spec:

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 2Gi

storageClassName: fast-storage

**Step 2: Apply the PVC**

kubectl apply -f pvc.yaml

**8️. Upgrading a Kubernetes Cluster**

**📝 Question:**  
Upgrade a Kubernetes cluster from **v1.25** to **v1.26**.

**🔹 Solution:**

**Step 1: Drain the control plane node**

kubectl drain master-node --ignore-daemonsets

**Step 2: Upgrade kubeadm**

sudo apt-get update && sudo apt-get install -y kubeadm=1.26.0-00

sudo kubeadm upgrade apply v1.26.0

**Step 3: Upgrade kubelet and kubectl**

sudo apt-get install -y kubelet=1.26.0-00 kubectl=1.26.0-00

sudo systemctl restart kubelet

**Step 4: Uncordon the node**

kubectl uncordon master-node

**Exam Tips**

**Use aliases & autocompletion:**

sh

CopyEdit

alias k=kubectl

complete -F \_\_start\_kubectl k

**Use kubectl explain for quick reference:**

sh

CopyEdit

kubectl explain pod.spec

**Generate YAML quickly using kubectl run --dry-run**

sh

CopyEdit

kubectl run nginx --image=nginx --dry-run=client -o yaml > pod.yaml

**Check logs & events for debugging:**

sh

CopyEdit

kubectl logs <pod>

kubectl describe pod <pod>