CKA How the Certified Kubernetes Administrator exam works, what each domain covers, and a workflow for the cluster tasks, from etcd backups to a broken kubelet.

CKA exam guide

8 sections

PREPARED FOR
You

GENERATED
27 November 2026

VERSION
v0.2.449

IN SHORT

The Certified Kubernetes Administrator (CKA) exam gives you a terminal, several clusters and a list of tasks, and checks whether each cluster ends up in the state the task asked for. Where the CKAD is about applications, the CKA is about the cluster itself: installing and upgrading it, backing up etcd, scheduling, storage, networking and, above all, finding out why something is broken.

1 The format

- **Online and proctored.** A proctor watches through your webcam and checks your room before you start.
- **About two hours** of hands-on tasks (15 to 20), graded on what the clusters look like at the end. The Linux Foundation does not publish how tasks are weighted or how a partly finished one scores, so finish what you can and check it.
- **Performance-based.** You change real clusters, including their control plane and compute nodes.
- **What changes over time:** the pass mark, the number of tasks, the Kubernetes version, the price and the retake policy. They are in the official CKA page and the candidate handbook. Check them when you book and again the week before.

2 The environment

A remote desktop with a terminal and a browser that opens the Kubernetes documentation (`kubernetes.io/docs`) and the few other sites the handbook allows. Your own notes are not available.

Each task is done on a designated host. The task's infobox gives the `ssh` command: run it first, do the task there, and `exit` back to the base system (hostname `base`) before the next task. Nested ssh is not supported, so never ssh from one task host to another. Node-level work often needs root:

```bash
ssh <host-from-the-task>
sudo -i        # when the task needs root
exit           # leave root
exit           # back to base
```

The task hosts have `kubectl` with the `k` alias and completion, `yq` , `curl` and `wget` ; `base` has none of them, so `kubectl: command not found` means you have not ssh'd yet. (From the Linux Foundation's important instructions for the CKA and CKAD, checked 20 Nov 2026.)

3 The domains and their weights

Domain	Weight	What it asks of you
Troubleshooting	30%	Broken Pods, Services, nodes and control plane components; logs, events, `kubelet`, `crictl`
Cluster Architecture, Installation and Configuration	25%	kubeadm installs and upgrades, etcd backup and restore, RBAC, Helm and Kustomize
Services and Networking	20%	Services, Ingress and Gateway API, NetworkPolicies, CoreDNS
Workloads and Scheduling	15%	Deployments and rollouts, ConfigMaps and Secrets, taints, tolerations, affinity, resource limits
Storage	10%	StorageClasses, PersistentVolumes and claims, access modes, reclaim policies

The weights come from the CNCF curriculum; the official page has the current version. Troubleshooting is almost a third of the marks: practice it most.

4 The first minute

```bash
alias k=kubectl
source <(kubectl completion bash)
complete -o default -F __start_kubectl k
export do="--dry-run=client -o yaml"
export now="--force --grace-period 0"
```

5 A workflow for each task

1. **Read the whole task** first. Note the cluster, the node, the namespace, every name and every path.
2. **Switch context**, and `ssh` to the node only when the task needs it.
3. **Generate YAML** with `$do` rather than writing it, then edit what the generator cannot.
4. **Check the result** the way the grader will: `k get`, `k describe`, and for a node, `systemctl status kubelet`.
5. **Move on** after five minutes without progress. Flag the task and come back.

6 Cluster tasks to have done many times

Back up and restore etcd. Take the certificate paths from the etcd Pod's manifest, never from memory:

```bash
sudo -i
grep -E 'cert-file|key-file|trusted-ca-file|listen-client-urls'
  /etc/kubernetes/manifests/etcd.yaml
ETCDCTL_API=3 etcdctl snapshot save /opt/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
```

Upgrade with kubeadm. The control plane first, then each compute node; drain before and uncordon after:

```bash
kubeadm upgrade plan
kubeadm upgrade apply v1.x.y          # on the control plane
kubectl drain <node> --ignore-daemonsets
kubeadm upgrade node                  # on each compute node
systemctl restart kubelet
kubectl uncordon <node>
```

The exact package commands depend on the version; the Kubernetes docs page "Upgrading kubeadm clusters" has them, and it is allowed.

A node that is NotReady. Start with the kubelet:

```bash
systemctl status kubelet
journalctl -u kubelet --no-pager | tail -50
```

A wrong path or flag in `/var/lib/kubelet/config.yaml` or the kubelet's drop-in file is a classic. Fix it, then `systemctl daemon-reload && systemctl restart kubelet`.

A broken control plane component. Its manifest is in `/etc/kubernetes/manifests/`; the kubelet restarts the static Pod when the file changes. If `kubectl` itself stops answering, read the container's logs with `crictl ps -a` and `crictl logs <id>`.

7 Commands worth knowing by heart

```bash
k get nodes -o wide
k describe node <node> | grep -A5 Taints
k get events -A --sort-by=.lastTimestamp | tail
k auth can-i list pods --as=system:serviceaccount:<ns>:<sa>
k create role reader --verb=get,list --resource=pods $do
k create rolebinding reader --role=reader --serviceaccount=<ns>:<sa> $do
k get pv,pvc -A
k explain pod.spec.tolerations
```

8 How to practice

Work through the Gazeek CKA labs on a real kubeadm cluster: Learn shows every command, Practice gives you the goal and a nudge, Prove gives you only the goal, as the exam does. Break things on purpose and fix them; the troubleshooting marks come from having seen the failure before. The Memory Gym's CKA sets keep the facts fresh between labs.

From https://gazeek.com/resources/cka-exam-guide. For your own practice: the commands need a Kubernetes cluster to run against.