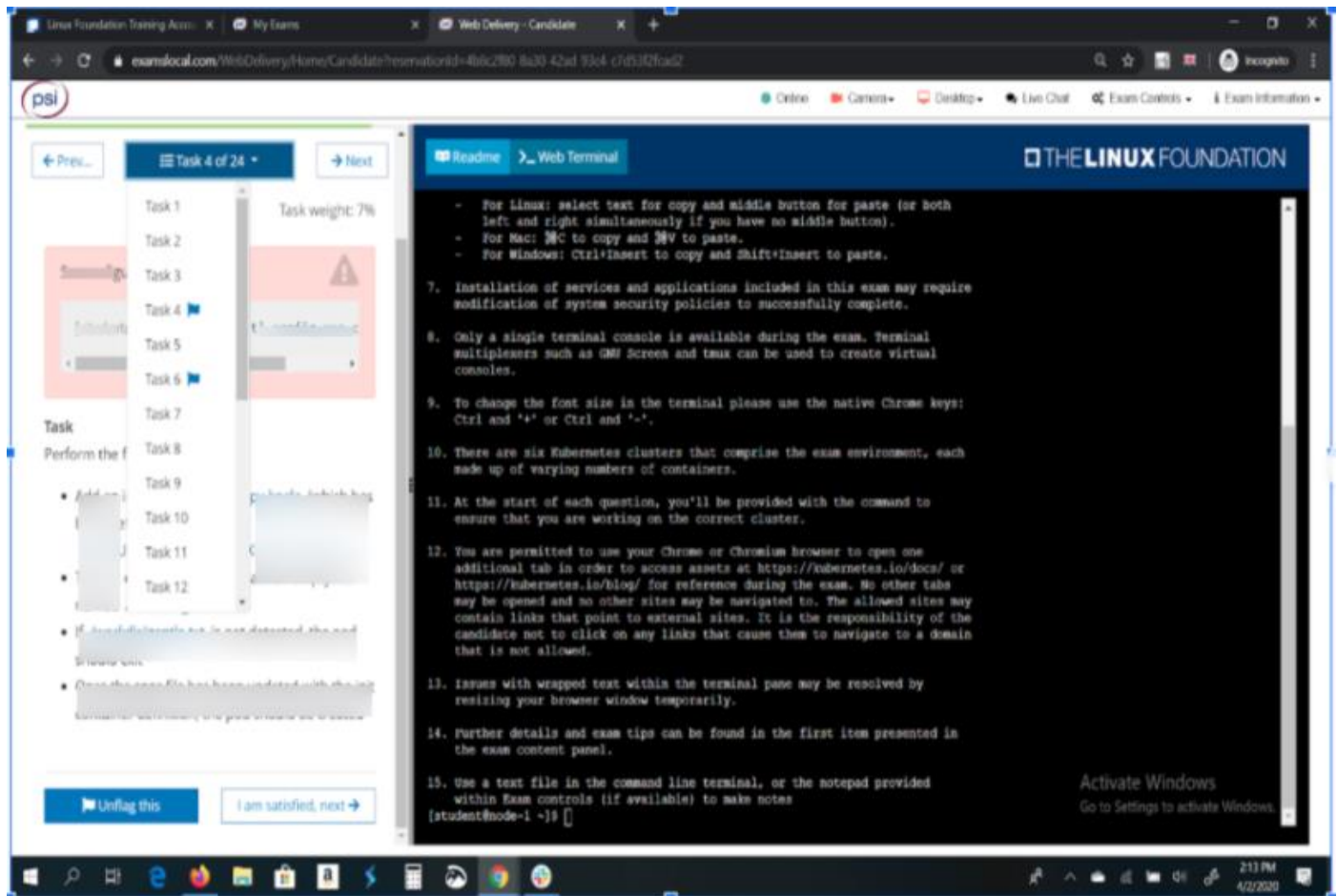


1 CKAD INSTRUCTION

Exam Portal UI:



Linux Foundation Certification Exams User Interface

<https://docs.linuxfoundation.org/tc-docs/certification/lf-candidate-handbook/exam-user-interface#exam-console-functions-in-top-menu-bar>

<https://docs.linuxfoundation.org/tc-docs/certification/lf-candidate-handbook/exam-user-interface#content-panel>

<https://docs.linuxfoundation.org/tc-docs/certification/lf-candidate-handbook/exam-user-interface#linux-server-terminal>

Each question will start with the context switching command like
kubectl config use-context k8s

Note: Use Kubernetes Official Documentation to Create Object Files. <https://kubernetes.io/> In Exam.

Note: This exam was built to give you a real exam like feel in terms of your ability to read and interpret a given question, validate your own work, manage time to complete given tasks.

Having said that:

1. Please note that this exam is not a replica of the actual exam.
2. Please note that the questions in this exam are not the same as in the actual exam.
3. Please note that the difficulty level may not be the same as in the actual exam.

Some Important points:

1. Don't panic and try to keep your cool.
2. Always complete the easy questions first. If something is impossible then attempt it when you have completed all other questions and rechecked all your answers.
3. Make Sure you run `kubectl config use-context <context>` before every question so you can perform task on correct cluster.
4. Practice creating resources using imperative commands e.g. **`kubectl create --dry-run -o yaml > file.yaml`**, `kubectl run`, `kubectl scale` and `kubectl expose` commands. This helped me to solve the easy questions much faster.
5. Take help from "`kubectl create -h`" command.
6. Try to use imperative commands as much as possible to save time.
7. Save your time for easy questions
8. Most of the Questions you have to run on Student-1 node. do not run commands on different clusters otherwise answers will not be registered.
9. You can bookmark below URLs for help:
 - <https://kubernetes.io/docs/reference/kubectl/cheatsheet/#viewing-finding-resources>
 - <https://kubernetes.io/docs/concepts/workloads/controllers/deployment/#creating-a-deployment>
 - <https://kubernetes.io/docs/tasks/configure-pod-container/static-pod/#static-pod-creation>
 - <https://kubernetes.io/docs/concepts/services-networking/service/>
 - <https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#step-two-add-a-nodeselector-field-to-your-pod-configuration>
 - <https://kubernetes.io/docs/concepts/scheduling-eviction/>

- <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/>
 - <https://kubernetes.io/docs/tasks/administer-cluster/kubeadm/kubeadm-upgrade/#upgrade-the-first-control-plane-node>
 - <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>
 - <https://kubernetes.io/docs/concepts/storage/persistent-volumes/#persistent-volumes>
 - <https://kubernetes.io/docs/concepts/storage/persistent-volumes/#persistentvolumeclaims>
 - <https://kubernetes.io/docs/concepts/storage/volumes/>
 - <https://kubernetes.io/docs/tasks/administer-cluster/configure-upgrade-etcd/#built-in-snapshot>
 - <https://kubernetes.io/docs/concepts/cluster-administration/logging/#streaming-sidecar-container>
 - <https://kubernetes.io/docs/reference/access-authn-authz/rbac/#clusterrole-example>
 - <https://kubernetes.io/docs/reference/access-authn-authz/rbac/#clusterrolebinding-example>
 - <https://kubernetes.io/docs/reference/access-authn-authz/rbac/#kubectl-create-clusterrole>
 - <https://kubernetes.io/docs/concepts/services-networking/network-policies/#networkpolicy-resource>
 - <https://kubernetes.io/docs/tasks/tls/managing-tls-in-a-cluster/>
 - <https://kubernetes.io/docs/concepts/services-networking/ingress/#the-ingress-resource>
10. Check *Bonus_Prepere_&_Register_For_CKA_&_CKAD_Exam_ed1* PPT from the Portal.



CKAD More Practice Questions and Notes Links:

- ckad-prep-notes
<https://github.com/twajr/ckad-prep-notes>
- CKAD Exercises
<https://github.com/dgkanatsios/CKAD-exercises>
- Practice Enough With These 150 Questions for the CKAD Exam
<https://medium.com/bb-tutorials-and-thoughts/practice-enough-with-these-questions-for-the-ckad-exam-2f42d1228552>

1. Time is very-very limited in exam so do not waste time in exam!!
2. Make Sure you go through all the videos and hands-on guides 2-3 times at least.
3. Make sure you perform all the practice questions available on the portal at least 4-5 times.
4. Make sure your desk and room are clean Proctor can ask you to move your camera multiple times if suspected any activity in the room.
5. Watch this Exam environment demo here: <https://www.youtube.com/watch?v=9UqkWcdy140>
6. Also Now you have 2 attempts for killer.sh after purchasing the CKA/CKAD/CKS certificate so make sure you use those attempts. Check here: <https://www.cncf.io/announcements/2021/06/02/linux-foundation-kubernetes-certifications-now-include-exam-simulator/>

Access from here

Exam Preparation Checklist

- ✓ Agree to Global Candidate Agreement [Read Now](#) ?
- ✓ Verify Name Status: Done [Verify Again](#) ?
- ✓ Select Platform Platform: Ubuntu 18.04 ?
- ☐ Schedule an Exam
Exam Code:
Exam Date: [Schedule](#)
[Click here to access the Exam Simulator](#) ?
- ✓ Check System Requirements Status: System Requirements Checked [Check Again](#) ?
- ✓ Get Candidate Handbook [Read Now](#) ?
- ✓ Read the Important Instructions [Read the Important Instructions](#) ?
- ☐ Take Exam Pending this section will be available once you have completed all the steps above

Where's my Free Retake? ?

Note:- These are the CKAD Mock questions only for the reference of what type of questions you might expect in the exam. In the reference question's answer, you might face an issue to replicate for this please check the answer in other Doc.

2 CKAD PRACTICE QUESTION

Q1) Create a secret and mount it as a volume.

Task:

- Create a secret mysecret with values user=myuser and password=myspassword
- Create an redis pod with volume redis-volume and mount secret as volume and put it on the path /etc/foo.

Q2) You are required to create a pod that requests a certain amount of CPU and memory, so it gets scheduled to a node that has those resources available.

- Create a pod named nginx-resources in the pod-resources namespace that requests a minimum of 200m CPU and 1Gi memory for its container
- The pod should use the nginx image

Q3) You are tasked to create a ConfigMap from a file.

Task

- Create an env file file.env with var1=val1
- Create a configmap envcfgmap from this env file and verify the configmap

Q4) Your application's namespace requires a specific service account to be used.

Task

- Create a namespace production
- Create a service account build-robot in production namespace
- Create a deployment app-a in production namespace with nginx image with number of replica 3.
- Update the app-a deployment in the production namespace to run as the restrictedservice service account.

Q5) A pod is running on the cluster but it is not responding.

Task

- The desired behavior is to have Kubernetes restart the pod when an endpoint returns an HTTP 500 on the /healthz endpoint. The service, probe-pod, should never send traffic to the pod while it is failing.

Please complete the following:

- The application has an endpoint, /started, that will indicate if it can accept traffic by returning an HTTP 200. If the endpoint returns an HTTP 500, the application has not yet finished initialization.
- The application has another endpoint /healthz that will indicate if the application is still working as expected by returning an HTTP 200. If the endpoint returns an HTTP 500 the application is no longer responsive.
- Configure the probe-pod pod provided to use these endpoints
- The probes should use port 8080

Q6) Task

Please complete the following;

- Deploy the counter pod to the cluster using the <https://raw.githubusercontent.com/kubernetes/website/master/content/en/examples/deploy/counter-pod.yaml> YAML spec file.
- Create a file /opt/log_output.txt
- Retrieve all currently available application logs from the running pod and store them in the file /opt/log_Output.txt.

Q7) Task

- From the pods running in namespace cpu-stress, write the name only of the pod that is consuming the most CPU to file /opt/file/pod.txt.

Q8) Task

Create a new deployment for running.nginx with the following parameters:

- Create namespace kdpd00201
- Run the deployment in the kdpd00201 namespace.
- Name the deployment frontend and configure with 4 replicas
- Configure the pod with a container image of lfcncf/nginx:1.13.7

- Set an environment variable of NGINX_PORT=8080 and also expose that port for the container above

Q9) Task

Please complete the following:

- Create namespace kdpd00202.
- Create deployment app in namespace kdpd00202 with image lfcncf/nginx:1.12
- Update the app deployment in the kdpd00202 namespace with a maxSurge of 5% and a maxUnavailable of 2%
- Perform a rolling update of the app deployment, changing the lfcncf/nginx image version to 1.13
- Roll back the app deployment to the previous version

Q10) Task

- Create a deployment kdsn00101-deployment in namespace kdsn00101 with image nginx.
- Add the func=webFrontEnd key/value label to the pod template metadata to identify the pod for the service definition
- Have 4 replicas

Next, create and deploy a service in namespace kdsn00101 that accomplishes the following:

- Exposes the service on TCP port 8080
- is mapped to the pods defined by the specification of kdsn00101-deployment
- Is of type NodePort has a name of cherry

Q11) Task

Follow the steps below to create a pod that will start at a pre-determined time and which runs to completion only once each time it is started:

- Create a YAML formatted Kubernetes manifest at periodic.yaml that runs the following shell command: **date in a single busybox** container.
- The command should run every minute and must complete within **22 seconds** or be terminated by Kubernetes.
- The Cronjob name and container name should both be **hello**
- Create the resource in the above manifest and verify that the job executes successfully at least once

Q12) A deployment is falling on the cluster due to an incorrect image being specified. Locate the deployment, and fix the problem.

Q13) A user has reported an application is unreachable due to a failing livenessProbe.

Task

Perform the following tasks:

- Find the broken pod and store its name and namespace to /opt/KDOB00401/broken.txt in the format: <namespace>/<pod>
The output file has already been created.
- Store the associated error events to a file /opt/KDOB00401/error.txt, The output file has already been created. You will need to use the -o wide output specifier with your command
- Fix the issue.

Q14) A project that you are working on has a requirement for persistent data to be available.

Task

To facilitate this, perform the following tasks:

- Create a file at /opt/KDSP00101/data/index.html with the content Acct=Finance
- Create a PersistentVolume named **task-pv-volume** using hostPath and allocate **1Gi** to it, specifying that the volume is at **/opt/KDSP00101/data** on the cluster's node. The configuration should specify the access mode of **ReadWriteOnce**. It should define the StorageClass name **exam** for the PersistentVolume, which will be used to bind PersistentVolumeClaim requests to this PersistentVolume.
- Create a PersistentVolumeClaim named **task-pv-claim** that requests a volume of at least **100Mi** and specifies an access mode of **ReadWriteOnce**
- Create a pod **myfrontend** that uses the PersistentVolumeClaim as a volume with a label **app: my-storage-app** mounting the resulting volume to a mountPath **/usr/share/nginx/html** inside the pod

Q15) Given a container that writes a log file in format A and a container that converts log files from format A to format B, create a deployment that runs both containers such that the log files from the first container are converted by the second container, emitting logs in format B.

Task:

- Create a deployment named deployment-xyz in the default namespace, that:
- Includes a primary `lfccncf/busybox:1` container, named `logger-dev`
- includes a sidecar `lfccncf/fluentd:v0.12` container, named `adapter-zen`
- Mounts a shared volume `/tmp/log` on both containers, which does not persist when the pod is deleted
- Instructs the `logger-dev` container to run the command

```
while true; do
  echo "I luv cncf">> /
  tmp/log/input.log;
  sleep 10;
done
```

which should output logs to `/tmp/log/input.log` in plain text format, with example values:

```
I luv cncf
I luv cncf
I luv cncf
```

- The `adapter-zen` sidecar container should read `/tmp/log/input.log` and output the data to `/tmp/log/output.*` in Fluentd JSON format.

Note: No knowledge of Fluentd is required to complete this task: all you will need to achieve this is to create the ConfigMap from the spec file provided at `/opt/test/fluentd-configmap.yaml`, and mount that ConfigMap to `/fluentd/etc` in the `adapter-zen` sidecar container.

Q16) Create a new pod called **super-user-pod** with image **busybox:1.28**. Allow the pod to be able to set `system_time`. **The container should sleep for 4800 seconds.**

Check more on: <https://kubernetes.io/docs/tasks/configure-pod-container/security-context/>

Q17) Create a deployment as follows

- Name `nginx-app`
- Using container `nginx` with version `1.11.10-alpine`
- The deployment should contain 3 replicas

Next, deploy the application with new version 1.13.0-alpine, by performing a rolling update, and record that update. Finally, rollback that update to the previous version 1.11.10-alpine

Q18) Create a new service account with the name **pvviewer**. Grant this Service account access to **list** all PersistentVolumes in the cluster by creating an appropriate cluster role called **pvviewer-role** and ClusterRoleBinding called **pvviewer-role-binding**.

Next, create a pod called **pvviewer** with the **image: redis** and **serviceaccount: pvviewer** in the default namespace.

Reference:

<https://kubernetes.io/docs/reference/access-authn-authz/rbac/#clusterrole-example>

<https://kubernetes.io/docs/reference/access-authn-authz/rbac/#kubectl-create-clusterrole>

<https://kubernetes.io/docs/reference/access-authn-authz/rbac/#clusterrolebinding-example>

Q19) Create a New NetworkPolicy named all-port that allows Pods in the existing namespace test-net to connect to port 80 of other Pods in same namespace.

Ensure that the new NetworkPolicy:

1. does not allow access to Pods not listening on port 80
2. does not allow access from Pods not in namespace test-net