

## CKAD Dumps

### Certified Kubernetes Application Developer (CKAD) Program

<https://www.certleader.com/CKAD-dumps.html>



#### NEW QUESTION 1

Exhibit:

Context

It is always useful to look at the resources your applications are consuming in a cluster. Task

- From the pods running in namespacecpu-stress , write the name only of the pod that is consuming the most CPU to file /opt/KDOBG030l/pod.txt, which has already been created.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

Solution:

#### NEW QUESTION 2

Exhibit:

Context

You sometimes need to observe a pod's logs, and write those logs to a file for further analysis. Task

Please complete the following;

- Deploy the counter pod to the cluster using the provided YAMLSpec file at /opt/KDOB00201/counter.yaml
- Retrieve all currently available application logs from the running pod and store them in the file /opt/KDOB00201/log\_Output.txt, which has already been created

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

Solution:

### NEW QUESTION 3

Exhibit:

Context

A user has reported an aopticaun is unteachable 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:

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.

- A. Mastered  
B. Not Mastered

**Answer:** A

**Explanation:**

Solution:

Create the Pod: `kubectl create -f`

`http://k8s.io/docs/tasks/configure-pod-container/`

`exec-liveness.yaml`

Within 30 seconds, view the Pod events: `kubectl describe pod liveness-exec`

The output indicates that no liveness probes have failed yet:

FirstSeen LastSeen CountFrom SubobjectPath Type Reason Message

-----

24s 24s 1{default-scheduler } Normal Scheduled Successfully assigned liveness-exec to worker0

23s 23s 1{kubelet worker0} spec.containers{liveness} Normal Pulling pulling image "gcr.io/google\_containers/busybox"

23s 23s 1{kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image "gcr.io/google\_containers/busybox"

23s 23s 1{kubelet worker0} spec.containers{liveness} Normal Created Created container with docker id 86849c15382e; Security:[seccomp=unconfined]

23s 23s 1{kubelet worker0} spec.containers{liveness} Normal Started Started container with docker id 86849c15382e

After 35 seconds, view the Pod events again: `kubectl describe pod liveness-exec`

At the bottom of the output, there are messages indicating that the liveness probes have failed, and the containers have been killed and recreated.

FirstSeen LastSeen Count From SubobjectPath Type Reason Message

-----

37s 37s 1{default-scheduler } Normal Scheduled Successfully assigned liveness-exec to worker0

36s 36s 1{kubelet worker0} spec.containers{liveness} Normal Pulling pulling image "gcr.io/google\_containers/busybox"

36s 36s 1{kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image "gcr.io/google\_containers/busybox"

36s 36s 1{kubelet worker0} spec.containers{liveness} Normal Created Created container with docker id 86849c15382e; Security:[seccomp=unconfined]

36s 36s 1{kubelet worker0} spec.containers{liveness} Normal Started Started container with docker id 86849c15382e

2s 2s 1{kubelet worker0} spec.containers{liveness} Warning Unhealthy Liveness probe failed: cat: can't open

`'/tmp/healthy': No such file or directory`

Wait another 30 seconds, and verify that the Container has been restarted: `kubectl get pod liveness-exec`

The output shows that RESTARTS has been incremented:

NAMEREADY STATUS RESTARTS AGE

liveness-exec 1/1 Running 1m

**NEW QUESTION 4**

Exhibit:

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 `lfcncf/busybox:1` container, named `logger-dev`
- Includes a sidecar `lfcncf/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

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

- The `adapter-zen` sidecar container should read `/tmp/log/input.log` and output the data to `/tmp/log/output.*` in Fluentd JSON format. Note that 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/KDMC00102/fluentd-configmap.yaml`, and mount that ConfigMap to `/fluentd/etc` in the `adapter-zen` sidecar container

- A. Mastered  
B. Not Mastered

**Answer:** A

**Explanation:**

Solution:

## NEW QUESTION 5

Exhibit:

### Context

A web application requires a specific version of redis to be used as a cache. Task

Create a pod with the following characteristics, and leave it running when complete:

- The pod must run in the web namespace. The namespace has already been created
- The name of the pod should be cache
- Use the Ifccncf/redis image with the3.2tag
- Expose port 6379

- A. Mastered
- B. Not Mastered

**Answer:** A

### Explanation:

Solution:

## NEW QUESTION 6

Exhibit:

Context

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

Task

Update the app-deployment in the production namespace to run as the restricted-service service account. The service account has already been created.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

Solution:

## NEW QUESTION 7

Exhibit:

Context

Developers occasionally need to submit pods that run periodically. Task

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

- Create a YAML formatted Kubernetes manifest /opt/KDPD00301/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

- A. Mastered  
B. Not Mastered

**Answer:** A

**Explanation:**  
Solution:

## NEW QUESTION 8

Context

Anytime a team needs to run a container on Kubernetes they will need to define a pod within which to run the container.

Task

Please complete the following:

- Create a YAML formatted pod manifest /opt/KDPD00101/pod1.yaml to create a pod named app1 that runs a container named app1cont using image lfcncf/arg-output with these command line arguments: -lines 56 -F
- Create the pod with the kubectl command using the YAML file created in the previous step
- When the pod is running display summary data about the pod in JSON format using the kubectl command and redirect the output to a file named /opt/KDPD00101/out1.json
- All of the files you need to work with have been created, empty, for your convenience

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

Solution:

**NEW QUESTION 10**

.....



## Thank You for Trying Our Product

\* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

\* One year free update

You can enjoy free update one year. 24x7 online support.

\* Trusted by Millions

We currently serve more than 30,000,000 customers.

\* Shop Securely

All transactions are protected by VeriSign!

**100% Pass Your CKAD Exam with Our Prep Materials Via below:**

<https://www.certleader.com/CKAD-dumps.html>