**CKAD Exam Questions**

[**Q1** 1](#_Toc80334770)

[**Q2** 3](#_Toc80334771)

[**Q3** 7](#_Toc80334772)

[**Q4** 11](#_Toc80334773)

[**Q5** 17](#_Toc80334774)

[**Q6** 19](#_Toc80334775)

[**Q7** 23](#_Toc80334776)

[**Q8** 27](#_Toc80334777)

[**Q9** 28](#_Toc80334778)

[**Q10** 34](#_Toc80334779)

[**Q11** 38](#_Toc80334780)

[**Q12** 42](#_Toc80334781)

[**Q13** 48](#_Toc80334782)

[**Q14** 52](#_Toc80334783)

[**Q15** 55](#_Toc80334784)

[**Q16** 57](#_Toc80334785)

[**Q17** 57](#_Toc80334786)

[**Q18** 58](#_Toc80334787)

[**Q19** 62](#_Toc80334788)

## **Q1**

Timeline

Description automatically generated with medium confidence

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 the 3.2 tag

\* Expose port 6379

**A**Solution:

A screenshot of a computer

Description automatically generated with medium confidence

**B**Solution:

A screenshot of a computer

Description automatically generated with medium confidence

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-ckad-topic-8-question-12-discussion)

**Suggested Answer:** *A*

## **Q2**

Graphical user interface, application

Description automatically generated

Context

You are tasked to create a secret and consume the secret in a pod using environment variables as follow:

Task

\* Create a secret named another-secret with a key/value pair; key1/value4

\* Start an nginx pod named nginx-secret using container image nginx, and add an environment variable exposing the value of the secret key key 1, using COOL\_VARIABLE as the name for the environment variable inside the pod

**A**Solution:

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

A screenshot of a computer

Description automatically generated

Text

Description automatically generated

**B**Solution:

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

A screenshot of a computer

Description automatically generated

Text

Description automatically generated

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-ckad-topic-9-question-11-discussion)

**Suggested Answer:** *B*

## **Q3**

Graphical user interface, application, chat or text message

Description automatically generated

Task

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

\* The pod-resources namespace has already been created

**A**Solution:

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

**B**Solution:

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

[Hide Solution](javascript:void();)

**Correct Answer:** *A*

## **Q4**

Graphical user interface, text, application, chat or text message

Description automatically generated

Context

You are tasked to create a ConfigMap and consume the ConfigMap in a pod using a volume mount.

Task

Please complete the following:

\* Create a ConfigMap named another-config containing the key/value pair: key4/value3

\* start a pod named nginx-configmap containing a single container using the

nginx image, and mount the key you just created into the pod under directory /also/a/path

**A**Solution:

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated

**B**Solution:

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

**Correct Answer:** *A*

## **Q5**

Graphical user interface, text, application, chat or text message

Description automatically generated

Context

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

Task

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

**A**Solution:

Text

Description automatically generated

**B**Solution:

Text

Description automatically generated

[Hide Solution](javascript:void();)    Discuss   **0**

**Correct Answer:** *A*

## **Q6**

Exhibit:

Graphical user interface, application

Description automatically generated

Context

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

Task

The desired behavior is to have Kubemetes 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

**A**Solution:

Text

Description automatically generated

In the configuration file, you can see that the Pod has a singleContainer. TheperiodSecondsfield specifies that the kubelet should perform a liveness probe every 5 seconds. TheinitialDelaySecondsfield tells the kubelet that it should wait 5 seconds before performing the first probe. To perform a probe, the kubelet executes the commandcat /tmp/healthyin the target container. If the command succeeds, it returns 0, and the kubelet considers the container to be alive and healthy. If the command returns a non-zero value, the kubelet kills the container and restarts it.  
When the container starts, it executes this command:  
/bin/sh -c 'touch /tmp/healthy; sleep 30; rm -rf /tmp/healthy; sleep 600'  
For the first 30 seconds of the container's life, there is a/tmp/healthyfile. So during the first 30 seconds, the commandcat /tmp/healthyreturns a success code. After 30 seconds,cat /tmp/healthyreturns a failure code.  
Create the Pod:  
kubectl apply -f https://k8s.io/examples/pods/probe/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 Count From 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 'k8s.gcr.io/busybox'  
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image 'k8s.gcr.io/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 'k8s.gcr.io/busybox'  
36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully  
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 thatRESTARTShas been incremented:  
NAME READY STATUS RESTARTS AGE  
liveness-exec 1/1 Running 1 1m

**B**Solution:

Text

Description automatically generated

In the configuration file, you can see that the Pod has a singleContainer. TheperiodSecondsfield specifies that the kubelet should perform a liveness probe every 5 seconds. TheinitialDelaySecondsfield tells the kubelet that it should wait 5 seconds before performing the first probe. To perform a probe, the kubelet executes the commandcat /tmp/healthyin the target container. If the command succeeds, it returns 0, and the kubelet considers the container to be alive and healthy. If the command returns a non-zero value, the kubelet kills the container and restarts it.  
When the container starts, it executes this command:  
/bin/sh -c 'touch /tmp/healthy; sleep 30; rm -rf /tmp/healthy; sleep 600'  
For the first 30 seconds of the container's life, there is a/tmp/healthyfile. So during the first 30 seconds, the commandcat /tmp/healthyreturns a success code. After 30 seconds,cat /tmp/healthyreturns a failure code.  
Create the Pod:  
kubectl apply -f https://k8s.io/examples/pods/probe/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 Count From 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 'k8s.gcr.io/busybox'  
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image 'k8s.gcr.io/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 'k8s.gcr.io/busybox'  
36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image 'k8s.gcr.io/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 thatRESTARTShas been incremented:  
NAME READY STATUS RESTARTS AGE  
liveness-exec 1/1 Running 1 1m

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-ckad-topic-3-question-7-discussion)

**Suggested Answer:** *B*

## **Q7**

Graphical user interface, application

Description automatically generated

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/KDOB0020l/log\_Output.txt, which has already been created

ASolution:

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

BSolution:

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Show Answer

Answer : **A**

## **Q8**

Graphical user interface, text, application, chat or text message

Description automatically generated

Context

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

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/KDOBG030l/pod.txt, which has already been created.

ASolution:

A screenshot of a computer

Description automatically generated with medium confidence

BSolution:

A screenshot of a computer

Description automatically generated with medium confidence

Show Answer

Answer : **A**

## **Q9**

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/podl.yml to create a pod named app1 that runs a container named app1cont using image Ifccncf/arg-output

with these command line arguments: -lines 56 -F

\* Create the pod with the kubect1 command using the YAML file created in the previous step

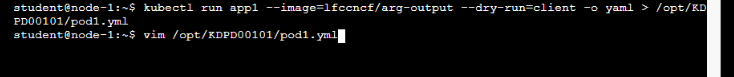
\* When the pod is running display summary data about the pod in JSON format using the kubect1 command and redirect the output to a file named /opt/KDPD00101/out1.json

\* All of the files you need to work with have been created, empty, for your convenience

Graphical user interface, text, application

Description automatically generated

ASolution:



Text

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated

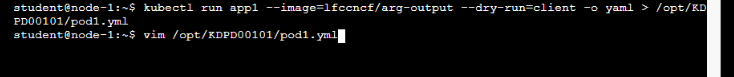
Text

Description automatically generated

Text

Description automatically generated

BSolution:



Text

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Show Answer

Answer : **A**

## **Q10**

Graphical user interface, text, application, chat or text message

Description automatically generated

Task

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

\* Run the deployment in the kdpd00201 namespace. The namespace has already been created

\* Name the deployment frontend and configure with 4 replicas

\* Configure the pod with a container image of lfccncf/nginx:1.13.7

\* Set an environment variable of NGINX\_\_PORT=8080 and also expose that port for the container above

ASolution:

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

BSolution:

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

Show Answer

Answer : **B**

## **Q11**

Exhibit:

Graphical user interface, application

Description automatically generated

Context

As a Kubernetes application developer you will often find yourself needing to update a running application.

Task

Please complete the following:

\* Update the app deployment in the kdpd00202 namespace with a maxSurge of 5% and a maxUnavailable of 2%

\* Perform a rolling update of the web1 deployment, changing the Ifccncf/ngmx image version to 1.13

\* Roll back the app deployment to the previous version

**A**Solution:

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface

Description automatically generated

**B**Solution:

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface

Description automatically generated

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-ckad-topic-9-question-1-discussion)

**Suggested Answer:** *A*

## **Q12**

**Timeline

Description automatically generated with medium confidence**

**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 Ifccncf/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**

**Text, chat or text message

Description automatically generated**

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

**Text

Description automatically generated with low confidence**

**\* 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-configma p.yaml , and mount that ConfigMap to /fluentd/etc in the adapter-zen sidecar container**

ASolution:

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

A screenshot of a computer

Description automatically generated

Text

Description automatically generated

A screen shot of a computer

Description automatically generated with low confidence

A screen shot of a computer

Description automatically generated with low confidence

BSolution:

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

A screenshot of a computer

Description automatically generated

Text

Description automatically generated

A screen shot of a computer

Description automatically generated with low confidence

## **Q13**

Graphical user interface, application

Description automatically generated

Context

You have been tasked with scaling an existing deployment for availability, and creating a service to expose the deployment within your infrastructure.

Task

Start with the deployment named kdsn00101-deployment which has already been deployed to the namespace kdsn00101 . Edit it to:

\* 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 ana deploy in namespace kdsn00l01 a service that accomplishes the following:

\* Exposes the service on TCP port 8080

\* is mapped to me pods defined by the specification of kdsn00l01-deployment

\* Is of type NodePort

\* Has a name of cherry

ASolution:

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

BSolution:

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Show Answer

Answer : **A**

## **Q14**

Graphical user interface, application

Description automatically generated

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 oy Kubernetes. The Cronjob namp 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**Solution:

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated

**B**Solution:

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-ckad-topic-4-question-4-discussion)

**Suggested Answer:** *A*

## **Q15**

Graphical user interface, application

Description automatically generated

Context

A container within the poller pod is hard-coded to connect the nginxsvc service on port 90 . As this port changes to 5050 an additional container needs to be added to the poller pod which adapts the container to connect to this new port. This should be realized as an ambassador container within the pod.

Task

\* Update the nginxsvc service to serve on port 5050.

\* Add an HAproxy container named haproxy bound to port 90 to the poller pod and deploy the enhanced pod. Use the image haproxy and inject the configuration located at /opt/KDMC00101/haproxy.cfg, with a ConfigMap named haproxy-config, mounted into the container so that haproxy.cfg is available at /usr/local/etc/haproxy/haproxy.cfg. Ensure that you update the args of the poller container to connect to localhost instead of nginxsvc so that the connection is correctly proxied to the new service endpoint. You must not modify the port of the endpoint in poller's args . The spec file used to create the initial poller pod is available in /opt/KDMC00101/poller.yaml

**A**Solution:  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
name: my-nginx  
spec:  
selector:  
matchLabels:  
run: my-nginx  
replicas: 2  
template:  
metadata:  
labels:  
run: my-nginx  
spec:  
containers:  
- name: my-nginx  
image: nginx  
ports:  
- containerPort: 90  
This makes it accessible from any node in your cluster. Check the nodes the Pod is running on:  
kubectl apply -f ./run-my-nginx.yaml  
kubectl get pods -l run=my-nginx -o wide  
NAME READY STATUS RESTARTS AGE IP NODE  
my-nginx-3800858182-jr4a2 1/1 Running 0 13s 10.244.3.4 kubernetes-minion-905m  
my-nginx-3800858182-kna2y 1/1 Running 0 13s 10.244.2.5 kubernetes-minion-ljyd  
Check your pods' IPs:  
kubectl get pods -l run=my-nginx -o yaml | grep podIP  
podIP: 10.244.3.4  
podIP: 10.244.2.5

**B**Solution:  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
name: my-nginx  
spec:  
selector:  
matchLabels:  
run: my-nginx  
- name: my-nginx  
image: nginx  
ports:  
- containerPort: 90  
This makes it accessible from any node in your cluster. Check the nodes the Pod is running on:  
kubectl apply -f ./run-my-nginx.yaml  
kubectl get pods -l run=my-nginx -o wide  
NAME READY STATUS RESTARTS AGE IP NODE  
my-nginx-3800858182-jr4a2 1/1 Running 0 13s 10.244.3.4 kubernetes-minion-905m  
my-nginx-3800858182-kna2y 1/1 Running 0 13s 10.244.2.5 kubernetes-minion-ljyd  
Check your pods' IPs:  
kubectl get pods -l run=my-nginx -o yaml | grep podIP  
podIP: 10.244.3.4  
podIP: 10.244.2.5

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-ckad-topic-2-question-3-discussion)

**Suggested Answer:** *A*

## **Q16**

Graphical user interface, application

Description automatically generated

Task

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

## **Q17**

Graphical user interface, application, chat or text message

Description automatically generated

Task

You have rolled out a new pod to your infrastructure and now you need to allow it to communicate with the web and storage pods but nothing else. Given the running pod kdsn00201 -newpod edit it to use a network policy that will allow it to send and receive traffic only to and from the web and storage pods.

Graphical user interface, text, application

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

## **Q18**

**Graphical user interface, application

Description automatically generated**

**Context**

**A user has reported an aopticauon 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:**

**A picture containing logo

Description automatically generated**

**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.**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

ASolution:  
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 Count From 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 thatRESTARTShas been incremented:  
NAME READY STATUS RESTARTS AGE  
liveness-exec 1/1 Running 1 m

BSolution:  
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 Count From 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'  
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 thatRESTARTShas been incremented:  
NAME READY STATUS RESTARTS AGE  
liveness-exec 1/1 Running 1 m

## **Q19**

Graphical user interface

Description automatically generated with medium confidence

Context

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 on node sk8s-node-0 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 PersistenetVolume.

\* Create a PefsissentVolumeClaim named task-pv-claim that requests a volume of at least 100Mi and specifies an access mode of ReadWriteOnce

\* Create a pod that uses the PersistentVolmeClaim as a volume with a label app: my-storage-app mounting the resulting volume to a mountPath /usr/share/nginx/html inside the pod

Graphical user interface, text, application, chat or text message

Description automatically generated

Text

Description automatically generated

**A**Solution:

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, monitor, black, screen

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

**B**Solution:

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, monitor, black, screen

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

[Hide Answer](https://www.pass4success.com/linux-foundation/discussions/exam-ckad-topic-8-question-6-discussion)

**Suggested Answer:** *A*