Linux-Foundation

Exam Questions CKAD

Certified Kubernetes Application Developer (CKAD) Program

NEW QUESTION 1
Exhibit:

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:

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: kubectlcreate-f
http://k8s.io/docs/tasks/configure-pod-container/
exec-liveness.yaml
Within 30 seconds, view the Pod events: kubectldescribepod liveness-exec
The output indicates that no liveness probes have failed yet:

```
FirstSeen LastSeen CountFrom SubobjectPath Type Reason Message
--------- -------- ----- ---- ------------- -------- ------ ------
24s 24s 1{default-scheduler } NormalScheduled Successfully assigned liveness-exec to worker0
23s 23s 1{kubelet worker0} spec.containers{liveness} NormalPulling pulling image"gcr.io/google_containers/busybox"
23s 23s 1{kubelet worker0} spec.containers{liveness} NormalPulled Successfully pulled image"gcr.io/google_containers/busybox"
23s 23s 1{kubelet worker0} spec.containers{liveness} NormalCreated Created container with docker id86849c15382e; Security:[seccomp=unconfined]
23s 23s 1{kubelet worker0} spec.containers{liveness} NormalStarted Started container with docker id86849c15382e
```

After 35 seconds, view the Pod events again: kubectldescribepod 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-exectoworker0
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 containerwithdocker id86849c15382e; Security:[seccomp=unconfined]
36s 36s 1{kubelet worker0} spec.containers{liveness} Normal Started Started containerwithdocker id86849c15382e
2s 2s 1{kubelet worker0} spec.containers{liveness} Warning Unhealthy Liveness probe failed: cat: can't open
'/tmp/healthy': No suchfileordirectory
```

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 STATUSRESTARTS AGE
liveness-exec 1/1Running 1m
```

NEW QUESTION 2
Exhibit:

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 of200m CPU and1Gi memory for its container
• The pod should use the nginx image

- The pod-resources namespace has already been created

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Solution:

NEW QUESTION 3
Exhibit:

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. Mastered
B. Not Mastered

Answer: A

Explanation:
Solution:

```
apiVersion:v1 kind:Pod metadata: labels: test:liveness
name:liveness-exec
spec: containers:
-name:liveness
image:k8s.gcr.io/busybox args:
- /bin/sh
- -c
- touch/tmp/healthy;sleep30;rm-rf/tmp/healthy;sleep600
livenessProbe: exec: command:
- cat
- /tmp/healthy initialDelaySeconds:5 periodSeconds:5
```

In the configuration file, you can see that the Pod has a single Container. The periodSeconds field specifies that the kubelet should perform a liveness probe every 5 seconds. The initialDelaySeconds field tells the kubelet that it should wait 5 seconds before performing the first probe. To perform a probe, the kubelet executes the command cat /tmp/healthy in 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/healthy file. So during the first 30 seconds, the command cat /tmp/healthy returns a success code. After 30 seconds, cat /tmp/healthy returns a failure co
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 that RESTARTS has been incremented: NAME READY STATUS RESTARTS AGE

```
liveness-exec 1/1 Running 1 1m
```

NEW QUESTION 4

Exhibit:

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, usingCOOL_VARIABLE as the name for the environment variable inside the pod

A. Mastered

B. Not Mastered

Answer: A

Explanation:

Solution:

NEW QUESTION 5

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

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

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-configma p.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 6

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 within22seconds 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. Mastered

B. Not Mastered

Answer: A

Explanation:

Solution:

NEW QUESTION 7

Exhibit:

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.

A. Mastered
B. Not Mastered

Answer: A

Explanation:

Suggest the Solution.

NEW QUESTION 8
Exhibit:

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=webFrontEndkey/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

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Solution:

NEW QUESTION 9

Exhibit:

Context

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

Please complete the following:

• Update theappdeployment in the kdpd00202 namespace with a maxSurge of5% and a maxUnavailable of2%

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

• Roll back theappdeployment to the previous version

A. Mastered

B. Not Mastered

Answer: A

Explanation:

Solution:

NEW QUESTION 10

......