Q1 - 4

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 in the web namespace. The namespace has already been created.
* The name of the pod should be cache.
* Use the lfccncf/redis image with the 4.0-alpine tag.
* Expose port 6379.

k run cache --image=lfccncf/redis:4.0-alpine --port=6379 $do > 1.yaml

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Q2 - 4

Context

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

Task

* Create a secret named another-secret with a key/value pair: key1/value4
* Start a nginx pod named nginx-secret using container image nginx, and add an environment variable exposing the value of the secret key key1 using BEST\_VARIABLE as the name of the environment variable inside the pod

kubectl create secret generic another-secret --from-literal=key1=value4 $do > /home/cloud\_user/ckad/2.yaml

A picture containing text, screenshot, monitor, screen

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

k run nginx-secret --image=nginx $do > /home/cloud\_user/ckad/2-nginxpod.yaml

<https://kubernetes.io/docs/tasks/inject-data-application/distribute-credentials-secure/#define-container-environment-variables-using-secret-data>

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Q3 - 4

Task

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

* Create pod named nginx-resources in the pod-resources namespace that request a minimum of 400m CPU and 2Gi memory for its container
* The pod should use the nginx image
* The pod-resources namespace has already been created

Text

Description automatically generated

k run nginx-resources --image=nginx --requests="cpu=400m,memory=2Gi" -n pod-resources $do > /home/cloud\_user/ckad/3.yaml

Flag --requests has been deprecated, has no effect and will be removed in the future.

<https://kubernetes.io/docs/tasks/configure-pod-container/assign-cpu-resource/>

Text

Description automatically generated

Text

Description automatically generated

Q4

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 some-config containing the key/value pair: ke4/value1
* 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 /some/path.

k create cm some-config --from-literal=ke4=value1

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

k run nginx-configmap --image=nginx $do > /home/cloud\_user/ckad/4-nginx-configmap.yaml

modify generated YAML to add configmap section as volume mount.

<https://kubernetes.io/docs/tasks/configure-pod-container/configure-pod-configmap/#populate-a-volume-with-data-stored-in-a-configmap>

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Q5.

Context

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

Task

Update the app-1 deployment in the frontend

namespace to run as the app-account1 service account. The service account has already been created.

Graphical user interface, text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

k -n frontend create deployment app-1 --image=nginx --replicas=2 ( You don’t need to create deployment in exam)

Text

Description automatically generated

A picture containing text, black, screen, silver

Description automatically generated

Text

Description automatically generated

Q6

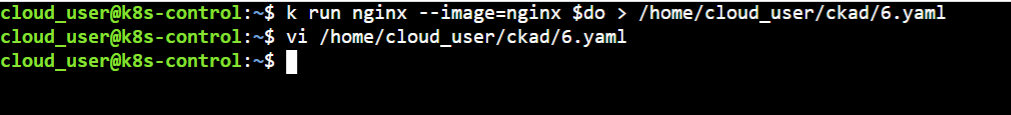
Context

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 return 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.



<https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-startup-probes/#define-a-liveness-http-request>

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Q7 - 4

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 YAML spec 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

Just I created 1 pod from to mimic the behaviour, in the exam just create pod using provided YAML.

<https://raw.githubusercontent.com/kubernetes/website/main/content/en/examples/admin/logging/two-files-counter-pod.yaml>

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

A computer screen capture

Description automatically generated with low confidence

Q8

Context

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

Task

From the pods running in namespace **stress** write the name **only** of the pod that is consuming the most CPU to file /opt/KDOB00201/pod.txt while it has already been created

Text

Description automatically generated

k -n stress top po --sort-by=cpu --use-protocol-buffers

echo stress > /home/cloud\_user/pod.txt

A screen shot of a computer

Description automatically generated with low confidence

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/pod1.yml to create a pod named app1 that runs a container named app1cont using image lfccncf/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 the files you need to work which have been created empty for your convenience

*Information: When creating your pod, you do not need to specify a container command, only args.*

Q10

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

* Run the deployment in the kdpd00201 namespace. The namespace has already created.
* Name the deployment frontend and configure with 6 replicas
* Configure the pod with a container image of lfccncf/nginx:1.13.7
* Set an environment variable of NGINX\_PORT=8001 and also expose that port for the container above

k -n kdpd00201 create deployment frontend --image=fccncf/nginx:1.13.7 --port=8001 $do >/home/cloud\_user/10.yaml

Text

Description automatically generated

kubectl explain pod.spec | grep -i "containers" -5

Q11

Context

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

Task

Please complete the following:

* Update the web1 deployment in the kdpd00202 namespace with a maxSurge of 10% and a maxUnavailable of 5%
* Perform a rolling update of the web1 deployment, changing the lfccncf/nginx image version to 1.13.8-alpine
* Roll back the web1 deployment to the previous version

K create ns kdpd00202

k -n kdpd00202 create deployment web1 --image=fccncf/nginx $do > 11.yaml

Text

Description automatically generated

Text

Description automatically generated

* Update the web1 deployment in the kdpd00202 namespace with a maxSurge of 10% and a maxUnavailable of 5% **using kubectl edit deployment**

Text

Description automatically generated

Text

Description automatically generated

Rolling update

Text

Description automatically generated

Rollback

k -n kdpd00202 rollout undo deployment web1 --to-revision=1

Text

Description automatically generated

Q12

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.

* Create a deployment named deployment-web in the default namespace that:
* Includes a primary lfccncf/busybox:1 container, named logger-dev
* Includes a sidecar lfccncf/fluentd:v0.12 container, named adaptor-123
* 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*

* The adaptor-123 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/KDMC00101/fluentd-configmap.yaml and mount that ConfigMap to /fluentd/etc in the adaptor-123 sidecar container

k create deployment deployment-web --image=lfccncf/busybox:1 --image=lfccncf/fluentd:v0.12 $do > /home/cloud\_user/ckad/12.yaml

Graphical user interface, text

Description automatically generated

Edit the YAML to update container names, mounts etc.

…. Still need to work on this

Q13

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

Task

Start with the deployment named kdns00101-deployment which has already been deployed to the namespace kdsn00101. Edit it to:

* Add the func=dmz key/value label to the pod template metadata to identify the pod for the service definition
* Have 4 replicas

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

* Exposes the service on TCP port 81
* Is mapped to the pods defined by the specification of kdsn00101-deployment
* Is of type NodePort
* Has a name of cherry

Ensure to update service yaml with func=dmz label

k create ns kdsn00101

k create deployment kdns00101-deployment --image=redis $do -n kdsn00101> 13.yaml

kcf /home/cloud\_user/ckad/13.yaml

A screenshot of a computer

Description automatically generated with medium confidence

before updating replicas and func:dmz label

Text

Description automatically generated with medium confidence

After update deployment with func: dmz label and replicas=4

Text

Description automatically generated

k -n kdsn00101 expose deployment kdns00101-deployment --name=cherry --port=81 --type=NodePort

Text

Description automatically generated

Q14

Context

Developers occasionally need to submit pods that run periodically.

Task

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

* Create a YAML formatted Kubernetes manifest /opt/KDPD00301/perodic.yaml that runs the following shell command: date in a singly busybox container. The command should run every minute and must complete within 17 seconds or be terminated by Kubernets. 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

k create cronjob hello --image=busybox --schedule="\*/1 \* \* \* \*" --restart=Never $do > /home/cloud\_user/ckad/14.yaml

update generated YAML with

activeDeadlineSeconds: 17

command: [/bin/sh, -c , “date”]

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Q15

Context

A container within the poller pod is hard-coded to connect the nginxsvc service on port 50. As this port changes to 6060 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 6060
* Add an HAproxy container named haproxy bound to port 50 to the poller pod and deploy the enhanced pod. Use the image haproxy and inject the configuration located at /opt/KDMC00101/haproxy.cfg which a ConfigMap named haproxy-config mounted in to the container so that haproxy.cfg is available at /usr/local/ext/haproxy/haproxy.cfg. Ensure that you update the args of the poller container to connect localhost intead 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

Need to update arg of poller pod from nginxsvc to localhost

Q18:

k get events -A | grep -i 'pulled image' | awk '{print $1 "/" $5 }' > /home/cloud\_user/ckad/awk.txt