⚠️ Catatan Penting

Semua sumber daya ini **hanya untuk penggunaan pribadi** dan **tidak boleh dibagikan** kepada pihak lain. Akses Anda terhubung dengan pendaftaran pelatihan Anda dan bertujuan menjaga kualitas serta eksklusivitas pengalaman belajar di Kubenesia.

CKAD Practice test

Topics related tracker:

1. Deployment, Service
2. Probes (liveness, readiness, startup)
3. Sidecar container + shared volume
4. Pod/Deployment custom serviceAccount
5. Pod/Deployment set environment variable
6. Pod Utilization / observability (kubectl top pods/nodes)
7. Run pod + expose specific container port
8. Job/Cronjob + Deadline seconds
9. Deployment incorrect image being specified, locate the deployment and fix the problem
10. Sidecar container
11. Storage
12. Container command + arguments
13. Kubectl log + export to a file as output
14. Resources requests & limits
15. Deployment maxSurge & maxUnavailable
16. ConfigMap as file / env
17. Secret as file / env
18. Troubleshoot, get events, export to a file

1. 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 and 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

2. 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

3. 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 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

4. Your application’s name space requires a specific service account to be used

   Task:
   Update the app-adeployment in the production namespace to run as the restrictedservice service account. The service account has already been created.

5. **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=8080and also expose that port for the container above

6. 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

7. 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 lfccncf/redis image with the3.2tag
   - Expose port 6379

8. 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

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

10. 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

11. 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 least100Mi 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

    You can access sk8s-node-0 by issuing the following command:

    ```
    [student@node-1] $ | ssh sk8s-node-0
    ```

    Ensure that you return to the base node (with hostname node-1) once you have compled your work on sk8s-node-0

12. 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 lfccncf/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

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

13. 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

14. **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

15. 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 lfccncf/ngmx image version to 1.13
- Roll back the app deployment to the previous version

16. 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

17. 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

18. 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.

The associated deployment could be running in any of the following namespace:

- qa
- test
- production
- alan

CKAD Practice Test

Answer:

- Create a Google Docs document, share document to team@kubenesia.com , training@kubenesia.com