

Linux-Foundation

Exam Questions CKAD

Certified Kubernetes Application Developer (CKAD) Program



NEW QUESTION 1

Exhibit:



Context

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

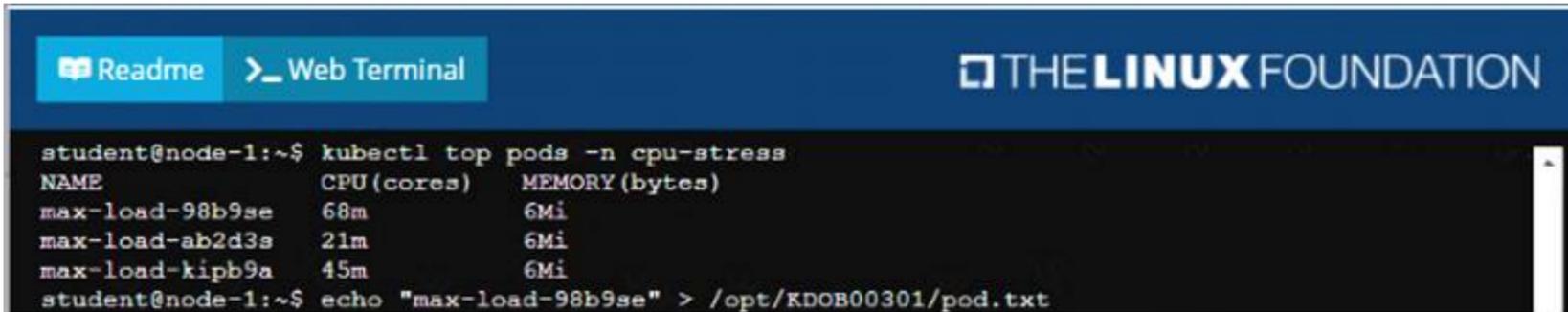
- From the pods running in namespacecpu-stress , write the name only of the pod that is consuming the most CPU to file /opt/KDOBG030I/pod.txt, which has already been created.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:



NEW QUESTION 2

Exhibit:



Context

A container within the poller pod is hard-coded to connect the nginxsvc service on port90 . As this port changes to5050 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 port5050.
- Add an HAProxy container named haproxy bound to port90 tothe 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. Mastered
- B. Not Mastered

Answer: A

Explanation:

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 -lrun=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 -lrun=my-nginx -o yaml | grep podIP podIP: 10.244.3.4

podIP: 10.244.2.5

NEW QUESTION 3

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.

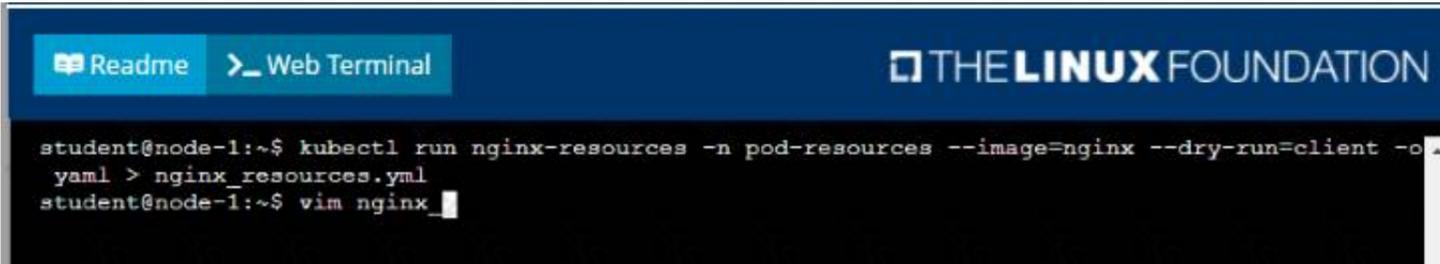
- Create a pod named nginx-resources in the pod-resources namespace that requests a minimum of 200m CPU and 1Gi memory for its container
- The pod should use the nginx image
- The pod-resources namespace has already been created

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:



```

Readme Web Terminal THE LINUX FOUNDATION
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx-resources
    name: nginx-resources
    namespace: pod-resources
spec:
  containers:
  - image: nginx
    name: nginx-resources
    resources:
      requests:
        cpu: 200m
        memory: "1Gi"
-- INSERT -- 15,22 All

```

```

Readme Web Terminal THE LINUX FOUNDATION
student@node-1:~$ kubectl run nginx-resources -n pod-resources --image=nginx --dry-run=client -o
yaml > nginx_resources.yml
student@node-1:~$ vim nginx_resources.yml
student@node-1:~$ kubectl create -g nginx_resources.yml
Error: unknown shorthand flag: 'g' in -g
See 'kubectl create --help' for usage.
student@node-1:~$ kubectl create -f nginx_resources.yml
pod/nginx-resources created
student@node-1:~$ kubectl get pods -n pod-re

```

```

Readme Web Terminal THE LINUX FOUNDATION
student@node-1:~$ kubectl get pods -n pod-resources
NAME          READY   STATUS    RESTARTS   AGE
nginx-resources 1/1     Running   0           8s
student@node-1:~$

```

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, using COOL_VARIABLE as the name for the environment variable inside the pod

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:


```

Readme Web Terminal THE LINUX FOUNDATION

student@node-1:~$ kubectl get pods -n web
NAME      READY   STATUS    RESTARTS   AGE
cache     1/1     Running   0           9s
student@node-1:~$ kubectl create secret generic some-secret --from-literal=key1=value4
secret/some-secret created
student@node-1:~$ kubectl get secret
NAME                TYPE          DATA   AGE
default-token-4kvr5  kubernetes.io/service-account-token  3       2d11h
some-secret          Opaque        1       5s
student@node-1:~$ kubectl run nginx-secret --image=nginx --dry-run=client -o yaml > nginx_secret.yml
student@node-1:~$ vim nginx_secret.yml
student@node-1:~$ kubectl create -f nginx_secret.yml
pod/nginx-secret created
student@node-1:~$ kubectl get pods
NAME          READY   STATUS             RESTARTS   AGE
liveness-http 1/1     Running            0           6h38m
nginx-101     1/1     Running            0           6h39m
nginx-secret  0/1     ContainerCreating  0           4s
poller        1/1     Running            0           6h39m
student@node-1:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
liveness-http 1/1     Running   0           6h38m
nginx-101     1/1     Running   0           6h39m
nginx-secret  1/1     Running   0           8s
poller        1/1     Running   0           6h39m
student@node-1:~$

```

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 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

```

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

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

Readme Web Terminal THE LINUX FOUNDATION

```
student@node-1:~$ kubectl create deployment deployment-xyz --image=lfcncf/busybox:1 --dry-run=c
lient -o yaml > deployment_xyz.yml
student@node-1:~$ vim deployment_xyz.yml
```

Readme Web Terminal THE LINUX FOUNDATION

```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: deployment-xyz
  name: deployment-xyz
spec:
  replicas: 1
  selector:
    matchLabels:
      app: deployment-xyz
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: deployment-xyz
    spec:
      containers:
      - image: lfcncf/busybox:1
        name: busybox
        resources: {}
status: {}
~
~
"deployment_xyz.yml" 24L, 434C 3,1 All
```

Readme Web Terminal THE LINUX FOUNDATION

```
kind: Deployment
metadata:
  labels:
    app: deployment-xyz
  name: deployment-xyz
spec:
  replicas: 1
  selector:
    matchLabels:
      app: deployment-xyz
  template:
    metadata:
      labels:
        app: deployment-xyz
    spec:
      volumes:
      - name: myvoll
        emptyDir: {}
      containers:
      - image: lfcncf/busybox:1
        name: logger-dev
        volumeMounts:
        - name: myvoll
          mountPath: /tmp/log
      - image: lfcncf/fluentd:v0.12
        name: adapterzen
3 lines yanked 27,22 Bot
```

Readme Web Terminal THE LINUX FOUNDATION

```

metadata:
  labels:
    app: deployment-xyz
spec:
  volumes:
  - name: myvol1
    emptyDir: {}
  - name: myvol2
    configMap:
      name: logconf
  containers:
  - image: lfccncf/busybox:1
    name: logger-dev
    command: ["/bin/sh", "-c", "while [ true ]; do echo 'i luv cncf' >> /tmp/log/input.log; sleep 10; done"]
    volumeMounts:
    - name: myvol1
      mountPath: /tmp/log
  - image: lfccncf/fluentd:v0.12
    name: adapter-zen
    command: ["/bin/sh", "-c", "tail -f /tmp/log/input.log >> /tmp/log/output.log"]
    volumeMounts:
    - name: myvol1
      mountPath: /tmp/log
    - name: myvol2
      mountPath: /fluentd/etc
  
```

37,33 Bot

```

student@node-1:~$ kubectl create -f deployment_xyz.yml
deployment.apps/deployment-xyz created
student@node-1:~$ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment-xyz 0/1     1             0           5s
student@node-1:~$ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment-xyz 0/1     1             0           9s
student@node-1:~$ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment-xyz 1/1     1             1           12s
student@node-1:~$ 
  
```

```

student@node-1:~$ kubectl create -f deployment_xyz.yml
deployment.apps/deployment-xyz created
student@node-1:~$ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment-xyz 0/1     1             0           5s
student@node-1:~$ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment-xyz 0/1     1             0           9s
student@node-1:~$ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment-xyz 1/1     1             1           12s
student@node-1:~$ 
  
```

NEW QUESTION 6

Exhibit:



Context

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

Task

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

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```

Readme Web Terminal THE LINUX FOUNDATION

student@node-1:~$ kubectl run cache --image=lfcncf/redis:3.2 --port=6379 -n web
pod/cache created
student@node-1:~$ kubectl get pods -n web
NAME      READY   STATUS             RESTARTS   AGE
cache     0/1     ContainerCreating  0          6s
student@node-1:~$ kubectl get pods -n web
NAME      READY   STATUS    RESTARTS   AGE
cache     1/1     Running   0          9s
student@node-1:~$

```

NEW QUESTION 7

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 a service in namespace `kdsn00101` a service that accomplishes the following:

- Exposes the service on TCP port 8080
- is mapped to the pods defined by the specification of `kdsn00101-deployment`
- Is of type `NodePort`
- Has a name of `cherry`

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```

student@node-1:~$ kubectl edit deployment kdsn00101-deployment -n kdsn00101

```

```

Readme Web Terminal THE LINUX FOUNDATION
Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
  creationTimestamp: "2020-10-09T08:50:39Z"
  generation: 1
  labels:
    app: nginx
  name: kdsn00101-deployment
  namespace: kdsn00101
  resourceVersion: "4786"
  selfLink: /apis/apps/v1/namespaces/kdsn00101/deployments/kdsn00101-deployment
  uid: 8d3ace00-7761-4189-ba10-fbc676c311bf
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
"/tmp/kubect1-edit-d4y5r.yaml" 70L, 1957C 1,1 Top

```

```

Readme Web Terminal THE LINUX FOUNDATION
uid: 8d3ace00-7761-4189-ba10-fbc676c311bf
spec:
  progressDeadlineSeconds: 600
  replicas: 4
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
        func: webFrontEnd
    spec:
      containers:
      - image: nginx:latest
        imagePullPolicy: Always
        name: nginx
        ports:
        - containerPort: 80

```

```

student@node-1:~$ kubectl edit deployment kdsn00101-deployment -n kdsn00101
deployment.apps/kdsn00101-deployment edited
student@node-1:~$ kubectl get deployment kdsn00101-deployment -n kdsn00101
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
kdsn00101-deployment 4/4      4             4           7h17m
student@node-1:~$ kubectl expose deployment kdsn00101-deployment -n kdsn00101 --type NodePort --
port 8080 --name cherry
service/cherry exposed

```

NEW QUESTION 8

Exhibit:



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 PersistentVolume.

- Create a PersistentVolumeClaim 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 PersistentVolumeClaim 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 completed your work on sk8s-node-0

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:
 Solution:

```
student@node-1:~$ kubectl config use-context sk8s
Switched to context "sk8s".
student@node-1:~$
```

```
* Documentation: https://help.ubuntu.com
* Management:   https://landscape.canonical.com
* Support:      https://ubuntu.com/advantage

System information as of Fri Oct 9 08:52:09 UTC 2020

System load: 2.02          Users logged in: 0
Usage of /: 10.3% of 242.29GB IP address for eth0: 10.250.3.115
Memory usage: 2%          IP address for docker0: 172.17.0.1
Swap usage: 0%            IP address for cni0: 10.244.1.1
Processes: 38

* Kubernetes 1.19 is out! Get it in one command with:

  sudo snap install microk8s --channel=1.19 --classic

https://microk8s.io/ has docs and details.

7 packages can be updated.
1 update is a security update.

New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

student@sk8s-node-0:~$
```

```
student@sk8s-node-0:~$ echo 'Acct=Finance' > /opt/KDSP00101/data/index.html
student@sk8s-node-0:~$ vim pv.yml
```


Readme Web Terminal THE LINUX FOUNDATION

```

apiVersion: v1
kind: Pod
metadata:
  name: mypod
  labels:
    app: my-storage-app
spec:
  containers:
  - name: myfrontend
    image: nginx
    volumeMounts:
    - mountPath: "/usr/share/nginx/html"
      name: mypod
  volumes:
  - name: mypod
    persistentVolumeClaim:
      claimName: task-pv-claim
  
```

17,32 All

```

student@sk8s-node-0:~$ kubectl create -f pod.yml
pod/mypod created
student@sk8s-node-0:~$ kubectl get
  
```

Readme Web Terminal THE LINUX FOUNDATION

```

student@sk8s-node-0:~$ kubectl get pods
NAME      READY   STATUS             RESTARTS   AGE
mypod    0/1     ContainerCreating   0           4s
student@sk8s-node-0:~$ kubectl get pods
NAME      READY   STATUS             RESTARTS   AGE
mypod    0/1     ContainerCreating   0           8s
student@sk8s-node-0:~$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
mypod    1/1     Running   0           10s
student@sk8s-node-0:~$ logout
Connection to 10.250.3.115 closed.
student@node-1:~$
  
```

NEW QUESTION 10

.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

CKAD Practice Exam Features:

- * CKAD Questions and Answers Updated Frequently
- * CKAD Practice Questions Verified by Expert Senior Certified Staff
- * CKAD Most Realistic Questions that Guarantee you a Pass on Your First Try
- * CKAD Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

100% Actual & Verified — Instant Download, Please Click
[Order The CKAD Practice Test Here](#)