

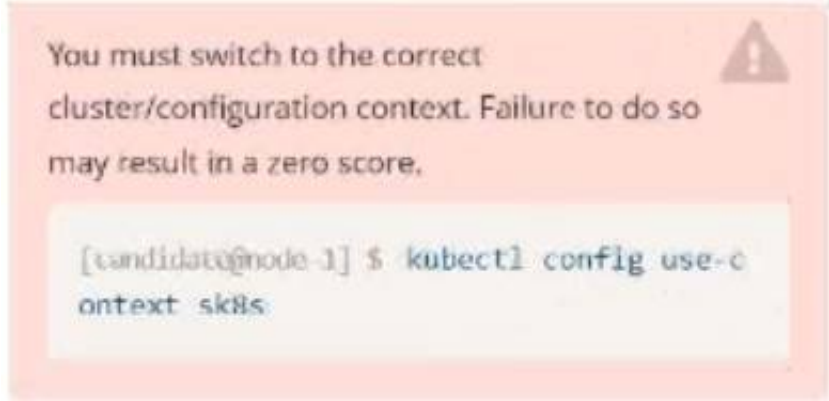
# Linux-Foundation

## Exam Questions CKAD

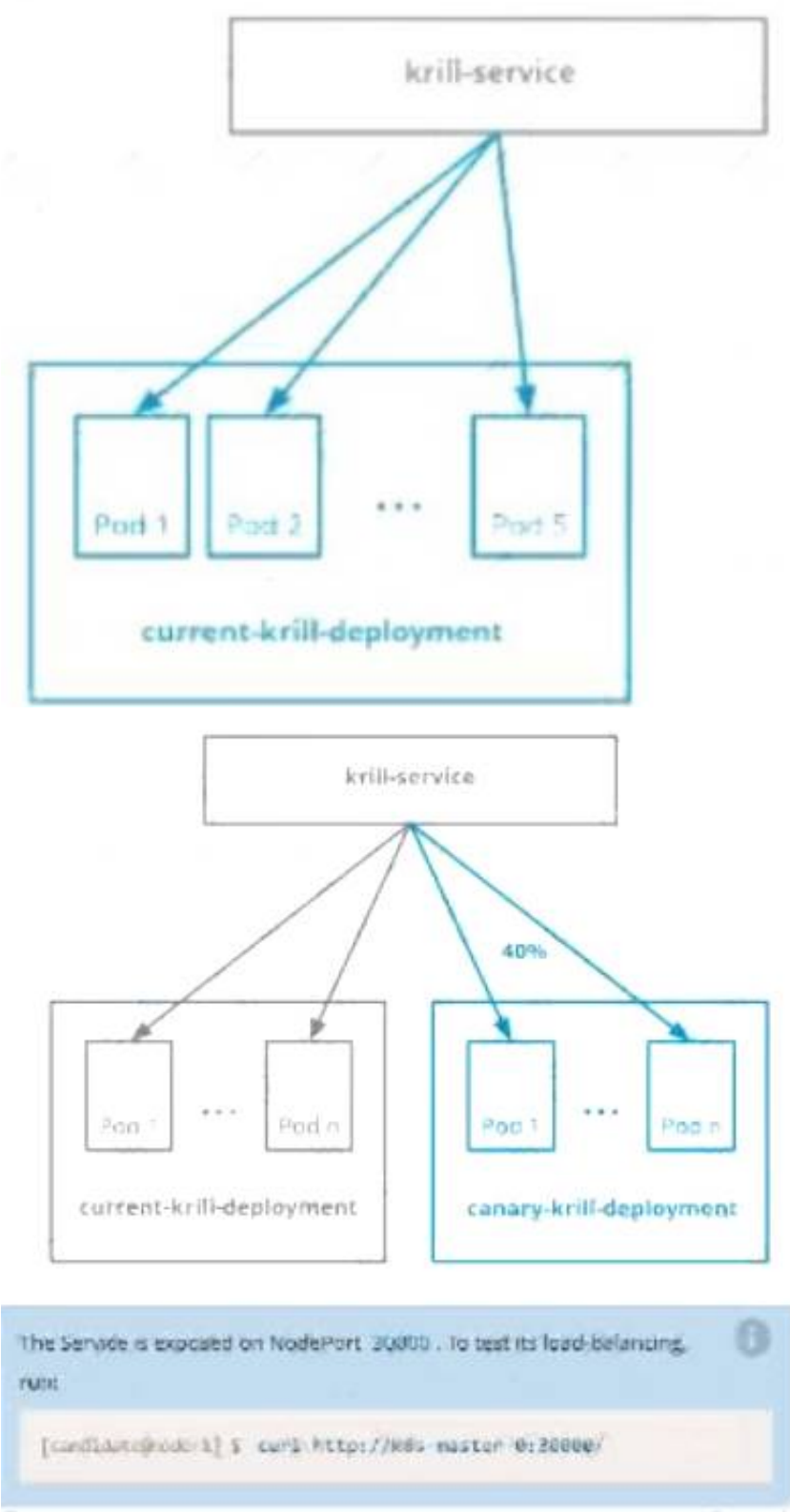
Certified Kubernetes Application Developer (CKAD) Program



NEW QUESTION 1  
Exhibit:



Context  
You are asked to prepare a Canary deployment for testing a new application release.  
Task:  
A Service named krill-Service in the goshawk namespace points to 5 pod created by the Deployment named current-krill-deployment



- A. Mastered
- B. Not Mastered

Answer: A

Explanation:  
Solution:

```
candidate@node-1:~/humane-stork$ kubectl scale deploy canary-krill-deployment --replicas 4 -n goshawk
deployment.apps/canary-krill-deployment scaled
candidate@node-1:~/humane-stork$ kubectl get deploy -n goshawk
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
canary-krill-deployment             4/4      4             4            46s
current-krill-deployment            5/5      5             5            7h22m
candidate@node-1:~/humane-stork$ wget https://k8s.io/examples/
```

Text Description automatically generated

```
File Edit View Terminal Tabs Help
candidate@node-1:~/humane-storks$ wget https://k8s.io/examples/admin/resource/quota-pod.yaml
--2022-09-24 11:43:51-- https://k8s.io/examples/admin/resource/quota-pod.yaml
Resolving k8s.io (k8s.io)... 34.107.204.206, 2600:1901:0:26f3::
Connecting to k8s.io (k8s.io)|34.107.204.206|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://kubernetes.io/examples/admin/resource/quota-pod.yaml [following]
--2022-09-24 11:43:52-- https://kubernetes.io/examples/admin/resource/quota-pod.yaml
Resolving kubernetes.io (kubernetes.io)... 147.75.40.148
Connecting to kubernetes.io (kubernetes.io)|147.75.40.148|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 90 [application/x-yaml]
Saving to: 'quota-pod.yaml'

quota-pod.yaml      100%[=====>]          90  --.-KB/s    in 0s

2022-09-24 11:43:52 (15.0 MB/s) - 'quota-pod.yaml' saved [90/90]

candidate@node-1:~/humane-storks$ vim quota-pod.yaml

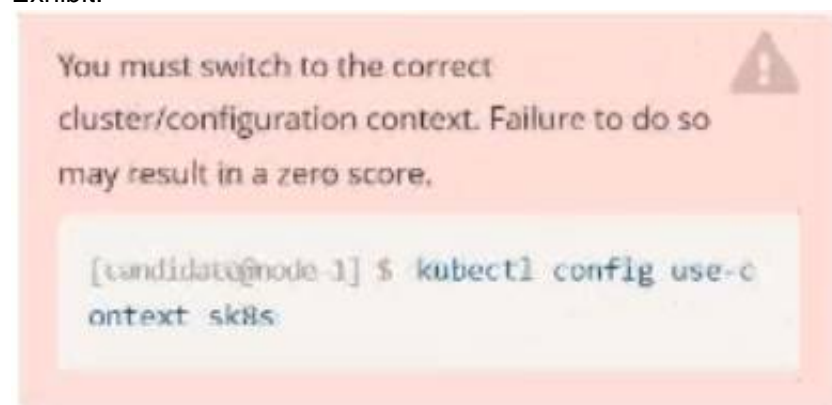
File Edit View Terminal Tabs Help

2022-09-24 11:43:52 (15.0 MB/s) - 'quota-pod.yaml' saved [90/90]

candidate@node-1:~/humane-storks$ vim quota-pod.yaml
candidate@node-1:~/humane-storks$ kubectl create -f quota-pod.yaml
resourcequota/pod-demo created
candidate@node-1:~/humane-storks$ kubectl get quota -n go
No resources found in go namespace.
candidate@node-1:~/humane-storks$ kubectl get quota -n goshawk
NAME      AGE      REQUEST    LIMIT
pod-demo  19s     pods: 9/10
candidate@node-1:~/humane-storks$ curl http://k8s-master-0:30000/
current-krill-deployment-fb7c7995c-kvtjr
app.kubernetes.io/name="current"
app.kubernetes.io/part-of="krill"
pod-template-hash="fb7c7995c"candidate@node-1:~/humane-storks$ curl http://k8s-master-0:30000/
current-krill-deployment-fb7c7995c-4whfm
app.kubernetes.io/name="current"
app.kubernetes.io/part-of="krill"
pod-template-hash="fb7c7995c"candidate@node-1:~/humane-storks$ curl http://k8s-master-0:30000/
canary-krill-deployment-5f78fd4786-dfk7l
app.kubernetes.io/name="canary"
app.kubernetes.io/part-of="krill"
pod-template-hash="5f78fd4786"candidate@node-1:~/humane-storks$ curl http://k8s-master-0:30000/
canary-krill-deployment-5f78fd4786-z5zrt
app.kubernetes.io/name="canary"
app.kubernetes.io/part-of="krill"
pod-template-hash="5f78fd4786"candidate@node-1:~/humane-storks$ curl http://k8s-master-0:30000/
canary-krill-deployment-5f78fd4786-2774b
app.kubernetes.io/name="canary"
app.kubernetes.io/part-of="krill"
pod-template-hash="5f78fd4786"candidate@node-1:~/humane-storks$
```

## NEW QUESTION 2

Exhibit:



Task:

Key3: value1

Add an environment variable named BEST\_VARIABLE consuming the value of the secret key3.

- A. Mastered
- B. Not Mastered

**Answer: A**

**Explanation:**

Solution:

```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl create secret generic app-secret -n default --from-literal=key3=value1
secret/app-secret created
candidate@node-1:~$ kubectl get secrets
NAME      TYPE      DATA   AGE
app-secret  Opaque    1       4s
candidate@node-1:~$ kubectl run nginx-secret -n default --image=nginx:stable --dry-run=client -o yaml > sec.yaml
candidate@node-1:~$ vim sec.yaml
```

Text Description automatically generated

```
File Edit View Terminal Tabs Help
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-secret
  name: nginx-secret
  namespace: default
spec:
  containers:
  - image: nginx:stable
    name: nginx-secret
    env:
    - name: BEST_VARIABLE
      valueFrom:
        secretKeyRef:
          name: app-secret
          key: key3
```

Text Description automatically generated

```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl create secret generic app-secret -n default --from-literal=key3=value1
secret/app-secret created
candidate@node-1:~$ kubectl get secrets
NAME          TYPE      DATA      AGE
app-secret    Opaque    1          4s
candidate@node-1:~$ kubectl run nginx-secret -n default --image=nginx:stable --dry-run=client -o yaml > sec.yaml
candidate@node-1:~$ vim sec.yaml
candidate@node-1:~$ kubectl create -f sec.yaml
pod/nginx-secret created
candidate@node-1:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nginx-secret  1/1     Running   0          7s
candidate@node-1:~$
```

### NEW QUESTION 3

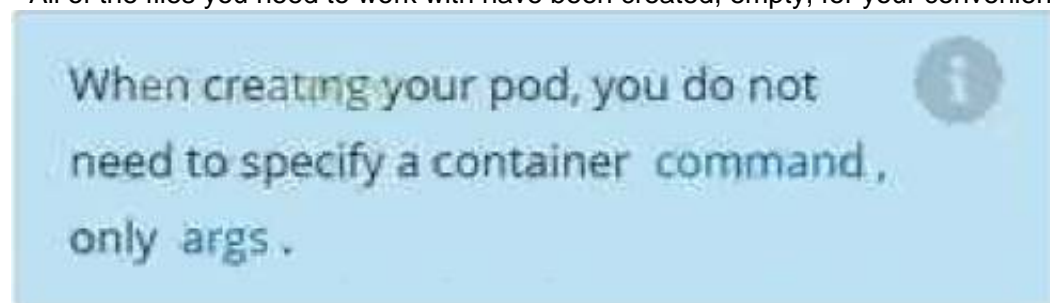
Context

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



- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```
student@node-1:~$ kubectl run app1 --image=lfccncf/arg-output --dry-run=client -o yaml > /opt/KD
PD00101/pod1.yml
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
```



3,1 All

11,30 All

```
pod/app1 created
student@node-1:~$ kubectl get pods
NAME                READY    STATUS              RESTARTS   AGE
app1                0/1     ContainerCreating   0           5s
counter             1/1     Running             0           4m44s
liveness-http       1/1     Running             0           6h50m
nginx-101           1/1     Running             0           6h51m
nginx-configmap     1/1     Running             0           6m21s
nginx-secret        1/1     Running             0           11m
poller              1/1     Running             0           6h51m
student@node-1:~$ kubectl get pods
NAME                READY    STATUS              RESTARTS   AGE
app1                1/1     Running             0           26s
counter             1/1     Running             0           5m5s
liveness-http       1/1     Running             0           6h50m
nginx-101           1/1     Running             0           6h51m
nginx-configmap     1/1     Running             0           6m42s
nginx-secret        1/1     Running             0           12m
poller              1/1     Running             0           6h51m
student@node-1:~$ kubectl delete pod app1
pod "app1" deleted
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
```

```
Readme Web Terminal

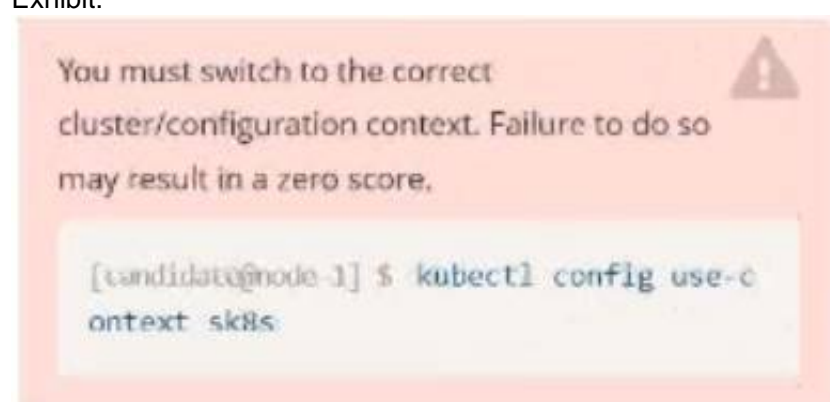
nginx-configmap 1/1 Running 0 6m2
nginx-secret 1/1 Running 0 11m
poller 1/1 Running 0 6h5
student@node-1:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
app1 1/1 Running 0 26s
counter 1/1 Running 0 5m5s
liveness-http 1/1 Running 0 6h50m
nginx-101 1/1 Running 0 6h51m
nginx-configmap 1/1 Running 0 6m42s
nginx-secret 1/1 Running 0 12m
poller 1/1 Running 0 6h51m
student@node-1:~$ kubectl delete pod app1
pod "app1" deleted
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
student@node-1:~$ kubectl create -f /opt/KDPD00101/pod1.yml
pod/app1 created
student@node-1:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
app1 1/1 Running 0 20s
counter 1/1 Running 0 6m57s
liveness-http 1/1 Running 0 6h52m
nginx-101 1/1 Running 0 6h53m
nginx-configmap 1/1 Running 0 8m34s
nginx-secret 1/1 Running 0 14m
poller 1/1 Running 0 6h53m
student@node-1:~$ kubectl get pod app1 -o json >
```

```
Readme Web Terminal THE LINUX FOUNDATION

poller 1/1 Running 0 6h51m
student@node-1:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
app1 1/1 Running 0 26s
counter 1/1 Running 0 5m5s
liveness-http 1/1 Running 0 6h50m
nginx-101 1/1 Running 0 6h51m
nginx-configmap 1/1 Running 0 6m42s
nginx-secret 1/1 Running 0 12m
poller 1/1 Running 0 6h51m
student@node-1:~$ kubectl delete pod app1
pod "app1" deleted
student@node-1:~$ vim /opt/KDPD00101/pod1.yml
student@node-1:~$ kubectl create -f /opt/KDPD00101/pod1.yml
pod/app1 created
student@node-1:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
app1 1/1 Running 0 20s
counter 1/1 Running 0 6m57s
liveness-http 1/1 Running 0 6h52m
nginx-101 1/1 Running 0 6h53m
nginx-configmap 1/1 Running 0 8m34s
nginx-secret 1/1 Running 0 14m
poller 1/1 Running 0 6h53m
student@node-1:~$ kubectl get pod app1 -o json > /opt/KDPD00101/out1.json
student@node-1:~$
student@node-1:~$
```

#### NEW QUESTION 4

Exhibit:



Task:

- > To run 2 replicas of the pod
- > Add the following label on the pod:

Role userUI

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

Text Description automatically generated

```
File Edit View Terminal Tabs Help
# reopened with the relevant failures.
#
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
  creationTimestamp: "2022-09-24T04:27:03Z"
  generation: 1
  labels:
    app: nginx
  name: ckad00017-deployment
  namespace: ckad00017
  resourceVersion: "3349"
  uid: 1cd67613-fade-46e9-b741-94298b9c6e7c
spec:
  progressDeadlineSeconds: 600
  replicas: 2
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
-- INSERT --
33,14 5%
```

Text Description automatically generated

```
File Edit View Terminal Tabs Help
name: ckad00017-deployment
namespace: ckad00017
resourceVersion: "3349"
uid: 1cd67613-fade-46e9-b741-94298b9c6e7c
spec:
  progressDeadlineSeconds: 600
  replicas: 2
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
        role: userUI
    spec:
      containers:
        - image: nginx:latest
          imagePullPolicy: Always
          name: nginx
          ports:
            - containerPort: 80
              protocol: TCP
          resources: {}
-- INSERT --
35,21 33%
```

Text Description automatically generated



```
File Edit View Terminal Tabs Help
backend-deployment-59d449b99d-h2zjq 0/1 Running 0 9s
backend-deployment-78976f74f5-b8c85 1/1 Running 0 6h40m
backend-deployment-78976f74f5-flfsj 1/1 Running 0 6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME READY UP-TO-DATE AVAILABLE AGE
backend-deployment 3/3 3 3 6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME READY UP-TO-DATE AVAILABLE AGE
backend-deployment 3/3 3 3 6h41m
candidate@node-1:~$ vim ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl set serviceaccount deploy app-1 app -n frontend
deployment.apps/app-1 serviceaccount updated
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ kubectl apply -f ~/prompt-escargot/buffalo-deployment.yaml
deployment.apps/buffalo-deployment configured
candidate@node-1:~$ kubectl get pods -n gorilla
NAME READY STATUS RESTARTS AGE
buffalo-deployment-776844df7f-r5fsb 1/1 Running 0 6h38m
buffalo-deployment-859898c6f5-zx5gj 0/1 ContainerCreating 0 8s
candidate@node-1:~$ kubectl get deploy -n gorilla
NAME READY UP-TO-DATE AVAILABLE AGE
buffalo-deployment 1/1 1 1 6h38m
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl edit deploy ckad00017-deployment -n ckad00017
deployment.apps/ckad00017-deployment edited
candidate@node-1:~$

File Edit View Terminal Tabs Help
candidate@node-1:~$ kubectl get pods -n gorilla
NAME READY STATUS RESTARTS AGE
buffalo-deployment-776844df7f-r5fsb 1/1 Running 0 6h38m
buffalo-deployment-859898c6f5-zx5gj 0/1 ContainerCreating 0 8s
candidate@node-1:~$ kubectl get deploy -n gorilla
NAME READY UP-TO-DATE AVAILABLE AGE
buffalo-deployment 1/1 1 1 6h38m
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl edit deploy ckad00017-deployment -n ckad00017
deployment.apps/ckad00017-deployment edited
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad0001
ckad00014 ckad00015 ckad00017
candidate@node-1:~$ kubectl expose deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
service/cherry exposed
candidate@node-1:~$

candidate@node-1:~$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 77d
candidate@node-1:~$ kubectl get svc -n ckad00017
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
cherry NodePort 10.100.100.176 <none> 8888:30683/TCP 24s
candidate@node-1:~$ kubectl expose service deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
Error from server (NotFound): services "deploy" not found
Error from server (NotFound): services "ckad00017-deployment" not found
candidate@node-1:~$ kubectl get svc -n ckad00017
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
cherry NodePort 10.100.100.176 <none> 8888:30683/TCP 46s
candidate@node-1:~$
```



```
File Edit View Terminal Tabs Help
candidate@node-1:~$ kubectl expose service deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
Error from server (NotFound): services "deploy" not found
Error from server (NotFound): services "ckad00017-deployment" not found
candidate@node-1:~$ kubectl get svc -n ckad00017
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
cherry    NodePort  10.100.100.176 <none>        8888:30683/TCP   46s
candidate@node-1:~$ history
 1 vi ~/spicy-pikachu/backend-deployment.yaml
 2 kubectl config use-context sk8s
 3 vim .vimrc
 4 vim ~/spicy-pikachu/backend-deployment.yaml
 5 kubectl apply -f ~/spicy-pikachu/backend-deployment.yaml
 6 kubectl get pods -n staging
 7 kubectl get deploy -n staging
 8 vim ~/spicy-pikachu/backend-deployment.yaml
 9 kubectl config use-context k8s
10 kubectl set serviceaccount deploy app-1 app -n frontend
11 kubectl config use-context k8s
12 vim ~/prompt-escargot/buffalo-deployment.yaml
13 kubectl apply -f ~/prompt-escargot/buffalo-deployment.yaml
14 kubectl get pods -n gorilla
15 kubectl get deploy -n gorilla
16 kubectl config use-context k8s
17 kubectl edit deploy ckad00017-deployment -n ckad00017
18 kubectl expose deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
19 kubectl get svc
20 kubectl get svc -n ckad00017
21 kubectl expose service deploy ckad00017-deployment -n ckad00017 --name=cherry --port=8888 --type=NodePort
22 kubectl get svc -n ckad00017
23 history
candidate@node-1:~$
```

NEW QUESTION 5

Exhibit:



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/KDOBG0301/pod.txt, which has already been created.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

```
Readme Web Terminal THE LINUX FOUNDATION
student@node-1:~$ kubectl top pods -n cpu-stress
NAME          CPU(cores)   MEMORY(bytes)
max-load-98b9se 68m          6Mi
max-load-ab2d3s 21m          6Mi
max-load-kipb9a 45m          6Mi
student@node-1:~$ echo "max-load-98b9se" > /opt/KDOBG00301/pod.txt
```

NEW QUESTION 6

Exhibit:



Task:

A pod within the Deployment named buffale-deployment and in namespace gorilla is logging errors.

1) Look at the logs identify errors messages.

Find errors, including User "system:serviceaccount:gorilla:default" cannot list resource "deployment" [...] in the namespace "gorilla"

The buffalo-deployment 'S manifest can be found at ~/prompt/escargot/buffalo-deployment.yaml

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

Text Description automatically generated

```
File Edit View Terminal Tabs Help
deployment.apps/backend-deployment configured
candidate@node-1:~$ kubectl get pods -n staging
NAME                                READY   STATUS    RESTARTS   AGE
backend-deployment-59d449b99d-cxct6 1/1     Running   0           20s
backend-deployment-59d449b99d-h2zjq 0/1     Running   0           9s
backend-deployment-78976f74f5-b8c85 1/1     Running   0          6h40m
backend-deployment-78976f74f5-flfsj 1/1     Running   0          6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
backend-deployment  3/3     3             3          6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
backend-deployment  3/3     3             3          6h41m
candidate@node-1:~$ vim ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl set serviceaccount deploy app-1 app -n frontend
deployment.apps/app-1 serviceaccount updated
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ vim ~/prompt-escargot/buffalo-deployment.yaml
candidate@node-1:~$ kubectl apply -f ~/prompt-escargot/buffalo-deployment.yaml
deployment.apps/buffalo-deployment configured
candidate@node-1:~$ kubectl get pods -n gorilla
NAME                                READY   STATUS              RESTARTS   AGE
buffalo-deployment-776844df7f-r5fsb 1/1     Running             0          6h38m
buffalo-deployment-859898c6f5-zx5gj 0/1     ContainerCreating   0           8s
candidate@node-1:~$ kubectl get deploy -n gorilla
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
buffalo-deployment  1/1     1             1          6h38m
candidate@node-1:~$
```

## NEW QUESTION 7

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
```

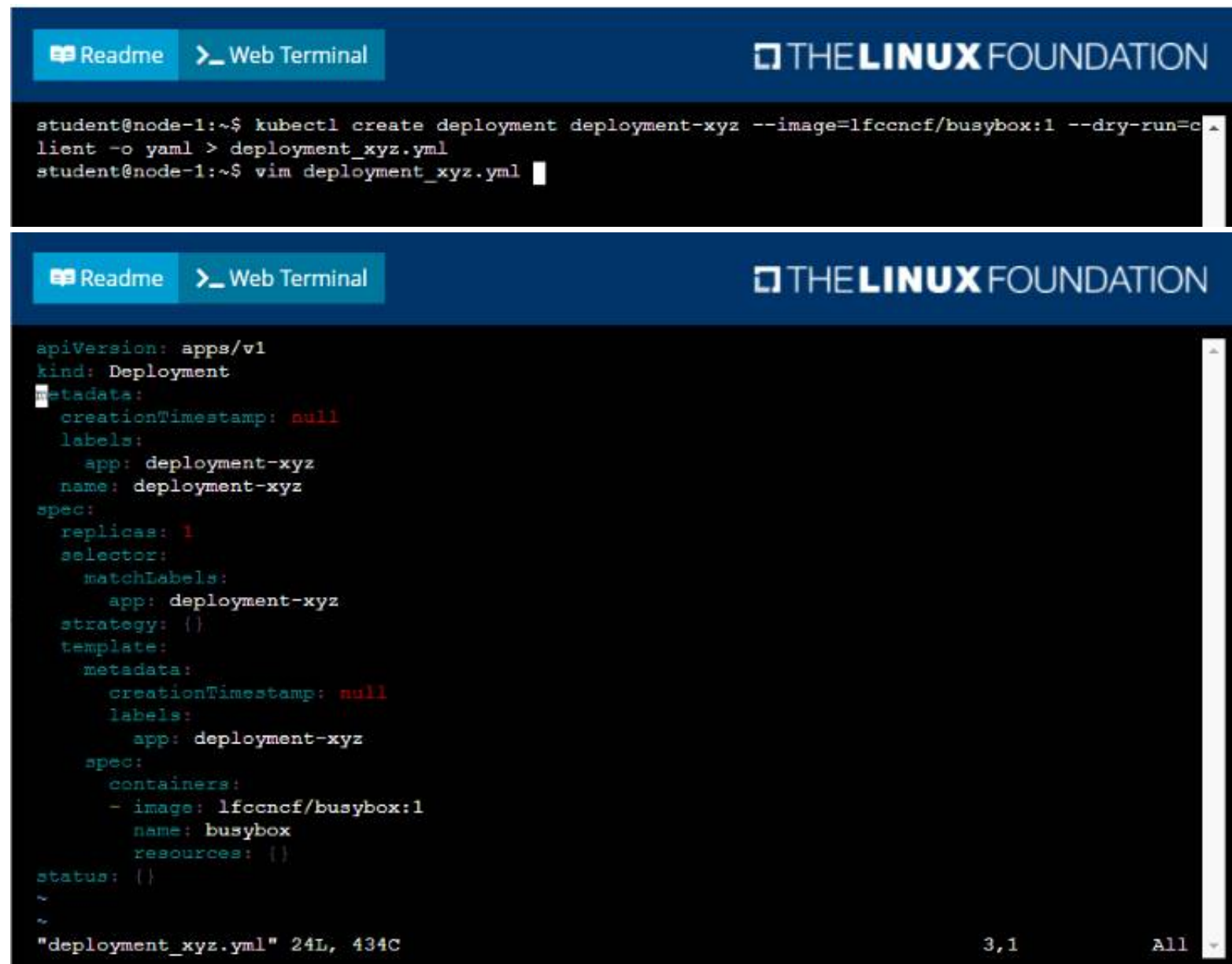
- The adapter-zen sidecar container should read /tmp/log/input.log and output the data to /tmp/log/output.\* in Fluentd JSON format. Note that no knowledge of Fluentd is required to complete this task: all you will need to achieve this is to create the ConfigMap from the spec file provided at /opt/KDMC00102/fluentd-configmap.p.yaml , and mount that ConfigMap to /fluentd/etc in the adapter-zen sidecar container

A. Mastered  
B. Not Mastered

**Answer:** A

**Explanation:**

Solution:



```
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

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: myvol1
        emptyDir: {}
      containers:
      - image: lfccncf/busybox:1
        name: logger-dev
        volumeMounts:
        - name: myvol1
          mountPath: /tmp/log
      - image: lfccncf/fluentd:v0.12
        name: adapter-zen
3 lines yanked
27,22
Bot

```

Readme
Web Terminal

THE LINUX FOUNDATION

```

replicas: 1
selector:
  matchLabels:
    app: deployment-xyz
template:
  metadata:
    labels:
      app: deployment-xyz
  spec:
    volumes:
    - name: myvol1
      emptyDir: {}
    containers:
    - image: lfccncf/busybox:1
      name: logger-dev
      command: ["/bin/sh", "-c", "while [ true ]; do echo 'i luv cnf' >> /tmp/log/input.log; sl
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
29,83
Bot

```

ReadmeWeb 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; sl
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
```

37,33Bot

```
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 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](#)**