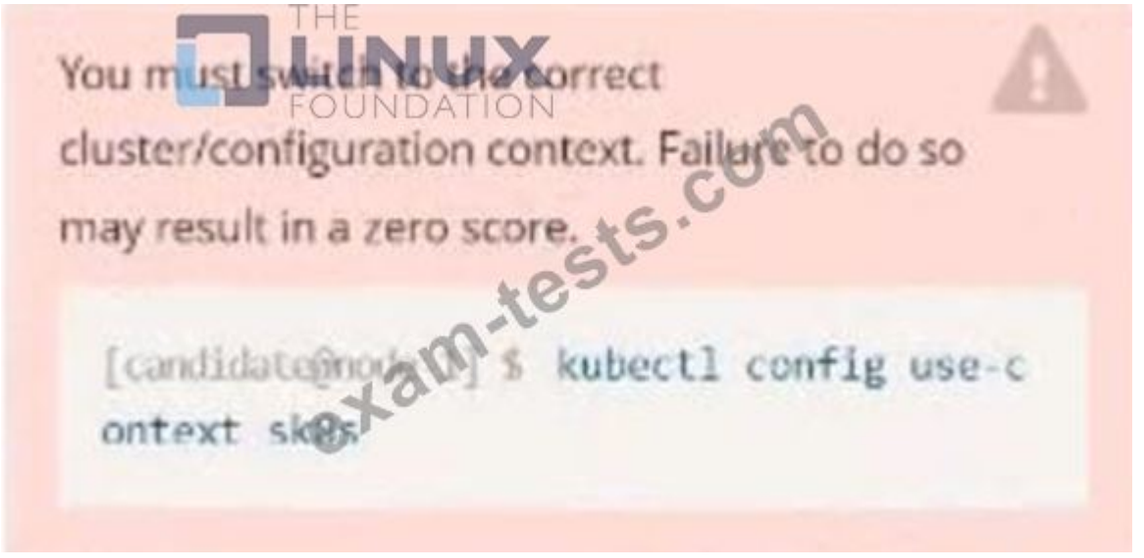
LinuxFoundation.CKAD.v2023-06-01.q33

Exam Code:	CKAD
Exam Name:	Linux Foundation Certified Kubernetes Application Developer Exam
Certification Provider:	Linux Foundation
Free Question Number:	33
Version:	v2023-06-01
# of views:	1540
# of Questions views:	330
https://www.exam-tests.com/CKAD-exam/LinuxFoundation.CKAD.v2023-06-01.q33.html	

NEW QUESTION: 1

Refer to Exhibit.



Task:

Update the Pod ckad00018-newpod in the ckad00018 namespace to use a NetworkPolicy allowing the Pod to send and receive traffic only to and from the pods web and db

Answer:

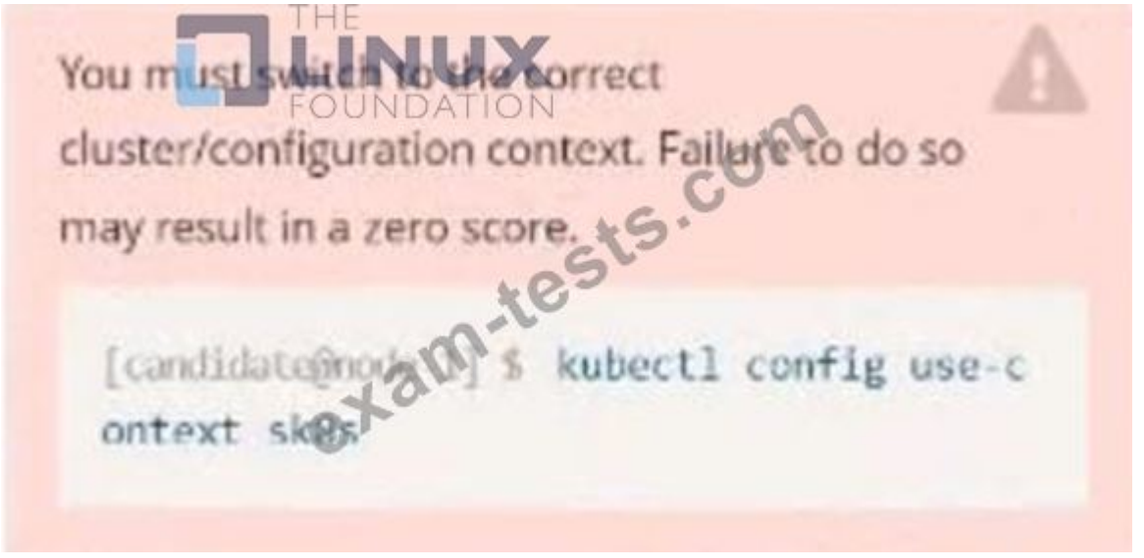
Solution:

```
e  Edit  View  Terminal  Tabs  Help
me:         all-access
mespace:    ckad00018
eated on:   2022-09-24 04:27:37 +0000 UTC
bels:       <none>
notations:  <none>
ec:
PodSelector:     all-access=true
Allowing ingress traffic:
  To Port: <any> (traffic allowed to all ports)
  From: <any> (traffic not restricted by source)
Allowing egress traffic:
  To Port: <any> (traffic allowed to all ports)
  To: <any> (traffic not restricted by destination)
Policy Types: Ingress, Egress

me:         default-deny
mespace:    ckad00018
eated on:   2022-09-24 04:27:37 +0000 UTC
bels:       <none>
notations:  <none>
ec:
PodSelector:     <none> (Allowing the specific traffic to all pods in this namespace)
Allowing ingress traffic:
  <none> (Selected pods are isolated for ingress connectivity)
Not affecting egress traffic
Policy Types: Ingress
ndidate@node-1:~$ kubectl  label  pod ckad00018-newpod -n ckad00018   web-access=true
d/ckad00018-newpod labeled
ndidate@node-1:~$ kubectl  label  pod ckad00018-newpod -n ckad00018   db-access=true
d/ckad00018-newpod labeled
```

NEW QUESTION: 2

Refer to Exhibit.



Task:

1) Fix any API depreciation issues in the manifest file -/credible-mite/www.yaml so that this application can be deployed on cluster K8s.

2) Deploy the application specified in the updated manifest file -/credible-mite/www.yaml in namespace cobra

Answer:

Solution:

NEW QUESTION: 3

Refer to Exhibit.

Context

You are tasked to create a ConfigMap and consume the ConfigMap in a pod using a volume mount.

Task

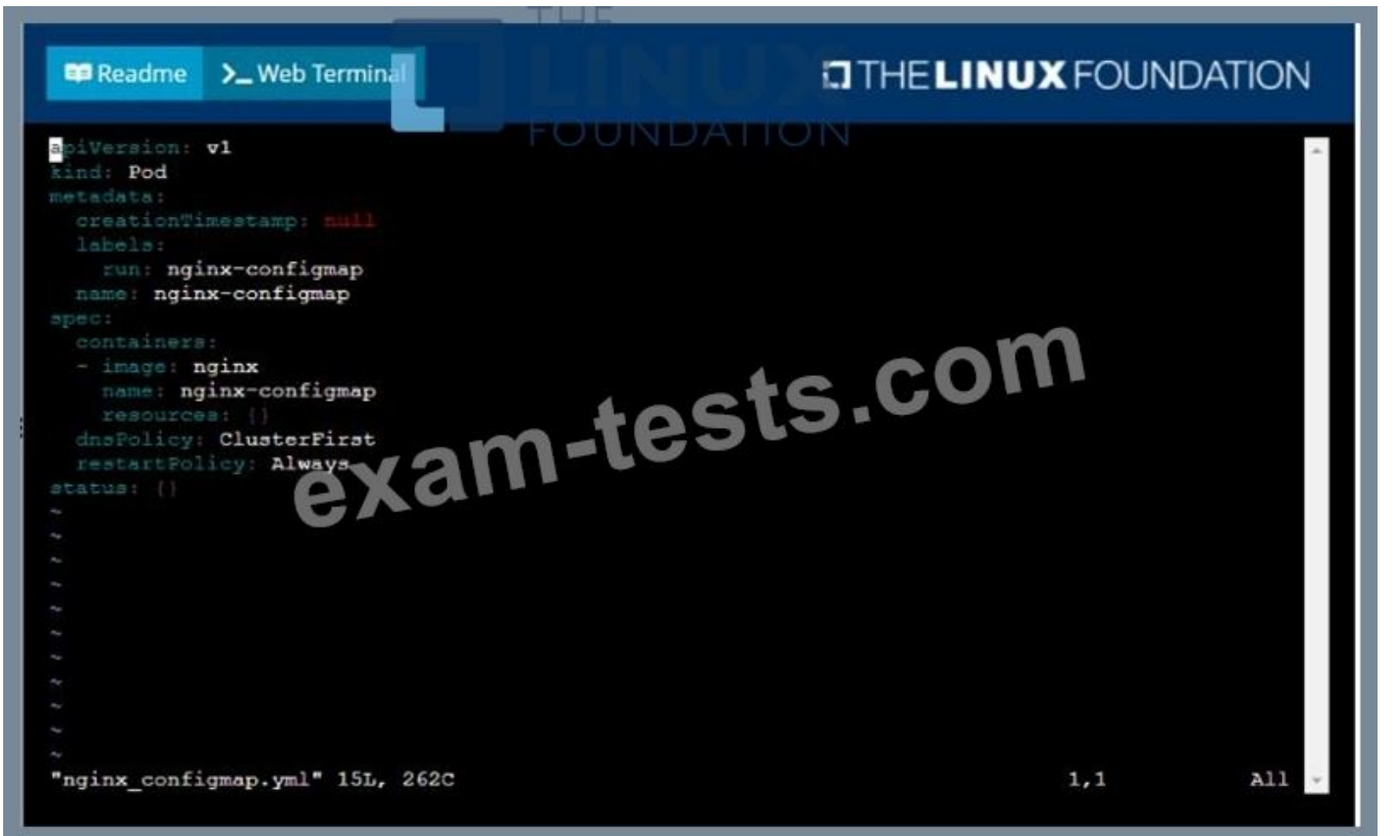
Please complete the following:

* Create a ConfigMap named 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

Answer:

Solution:



NEW QUESTION: 4

Context

Context

Developers occasionally need to submit pods that run periodically.

Task

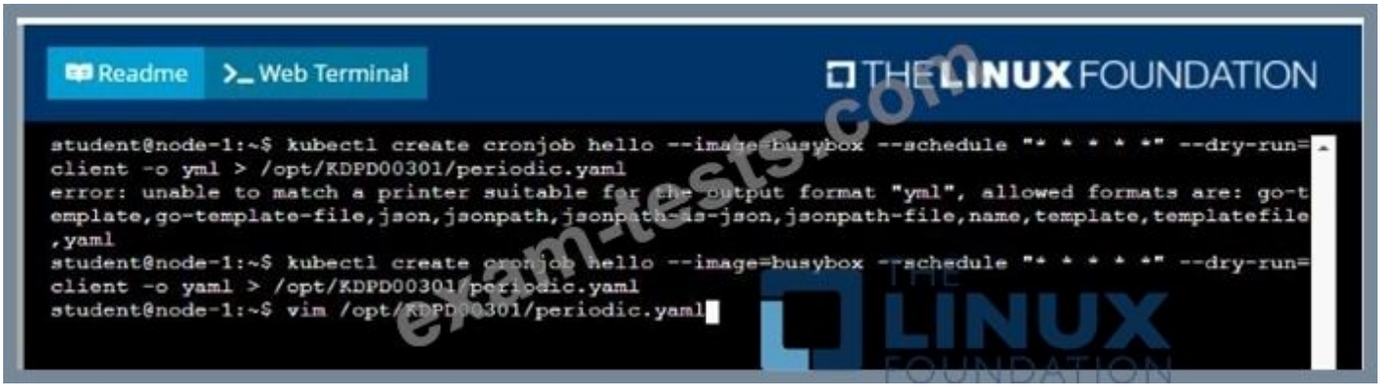
Follow the steps below to create a pod that will start at a predetermined time and]which runs to completion only once each time it is started:

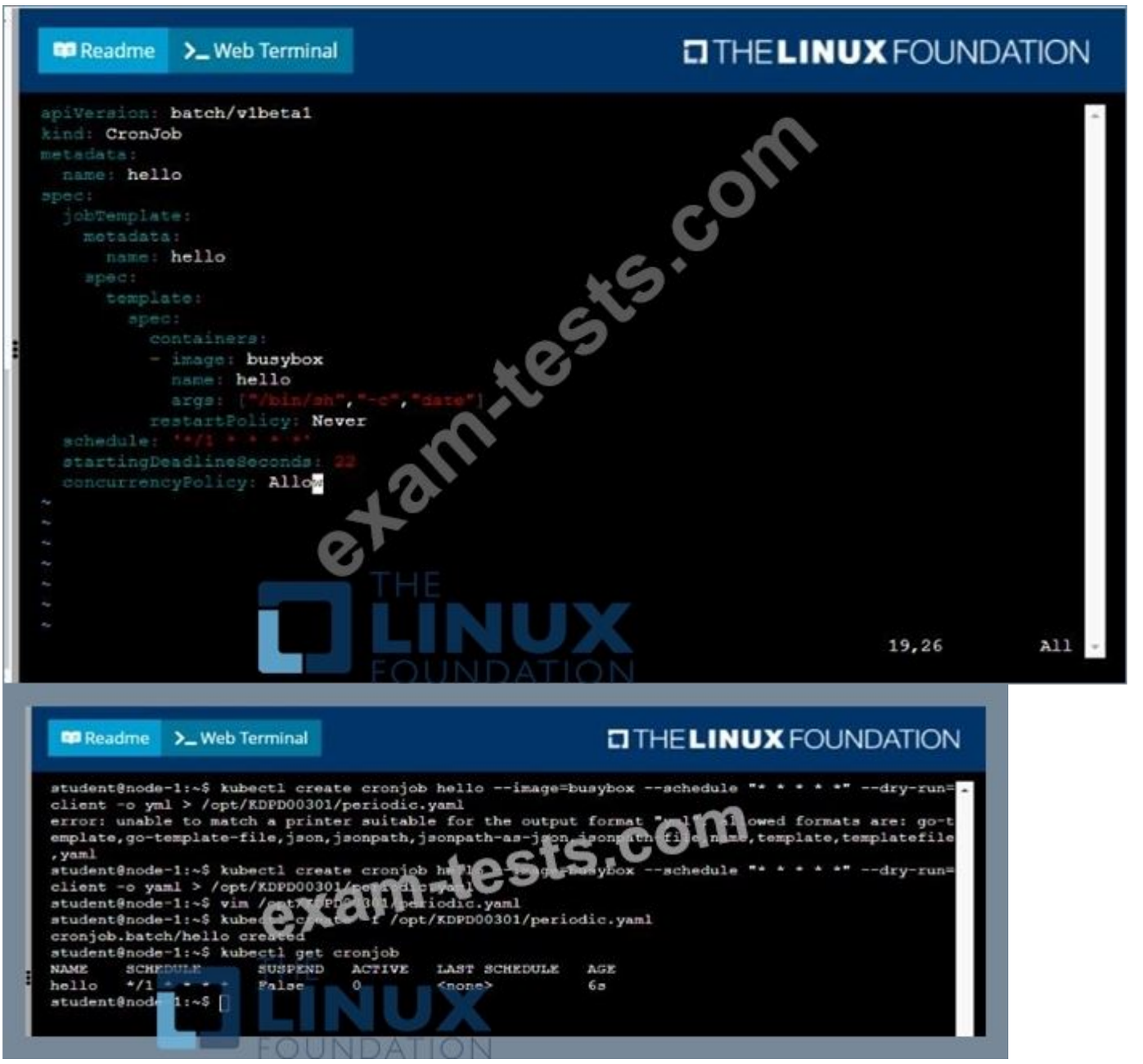
* Create a YAML formatted Kubernetes manifest /opt/KDPD00301/periodic.yaml that runs the following shell command: date in a single busybox container. The command should run every minute and must complete within 22 seconds or be terminated oy Kubernetes. The Cronjob namp and container name should both be hello

* Create the resource in the above manifest and verify that the job executes successfully at least once

Answer:

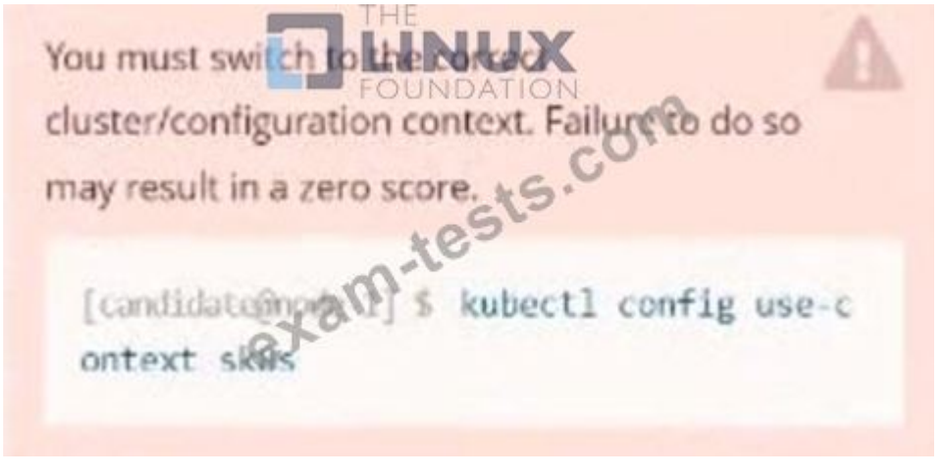
Solution:





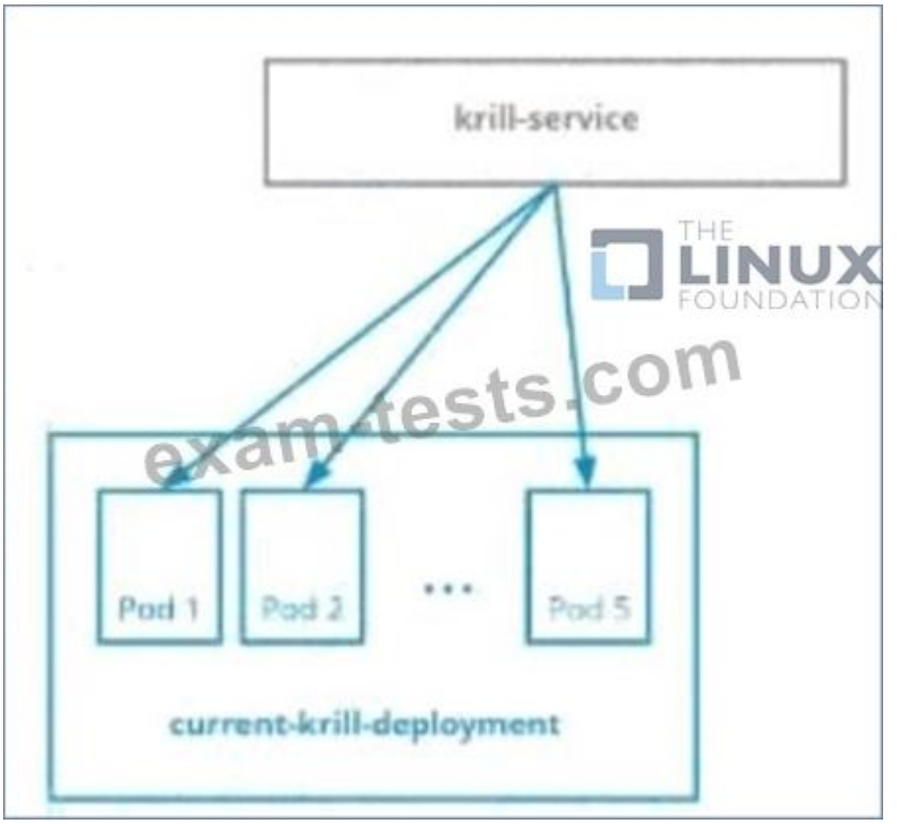
NEW QUESTION: 5

Context

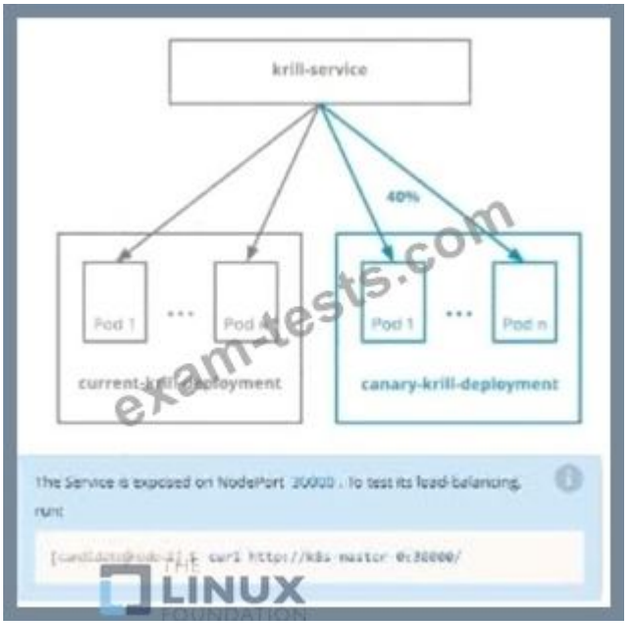


Context

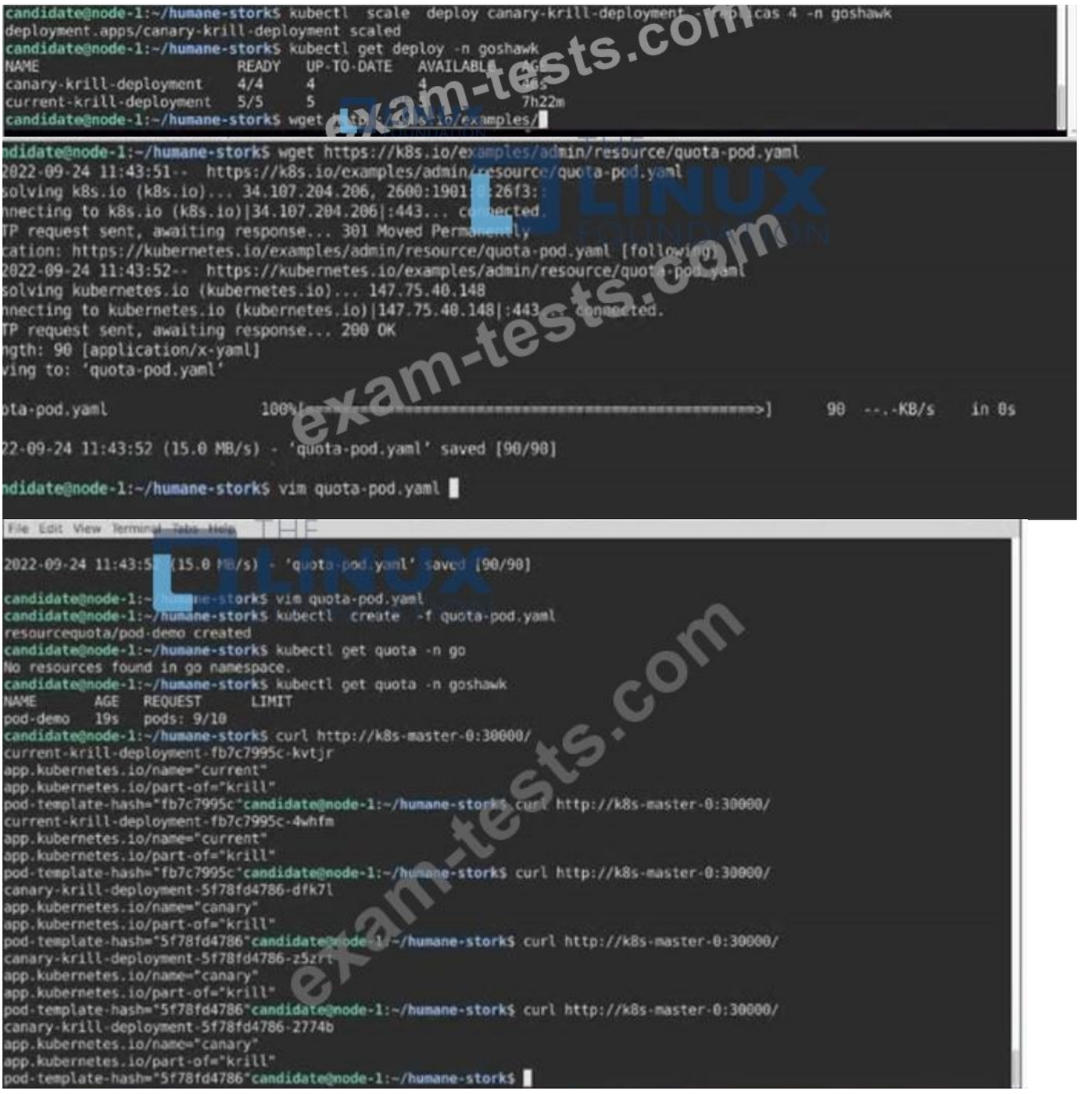
You are asked to prepare a Canary deployment for testing a new application release.
Task:
A Service named krill-Service in the goshark namespace points to 5 pod created by the Deployment named current-krill-deployment



1) Create an identical Deployment named canary-kill-deployment, in the same namespace.
2) Modify the Deployment so that:
-A maximum number of 10 pods run in the goshawk namespace.
-40% of the krill-service 's traffic goes to the canary-krill-deployment pod(s)



Answer:
Solution:



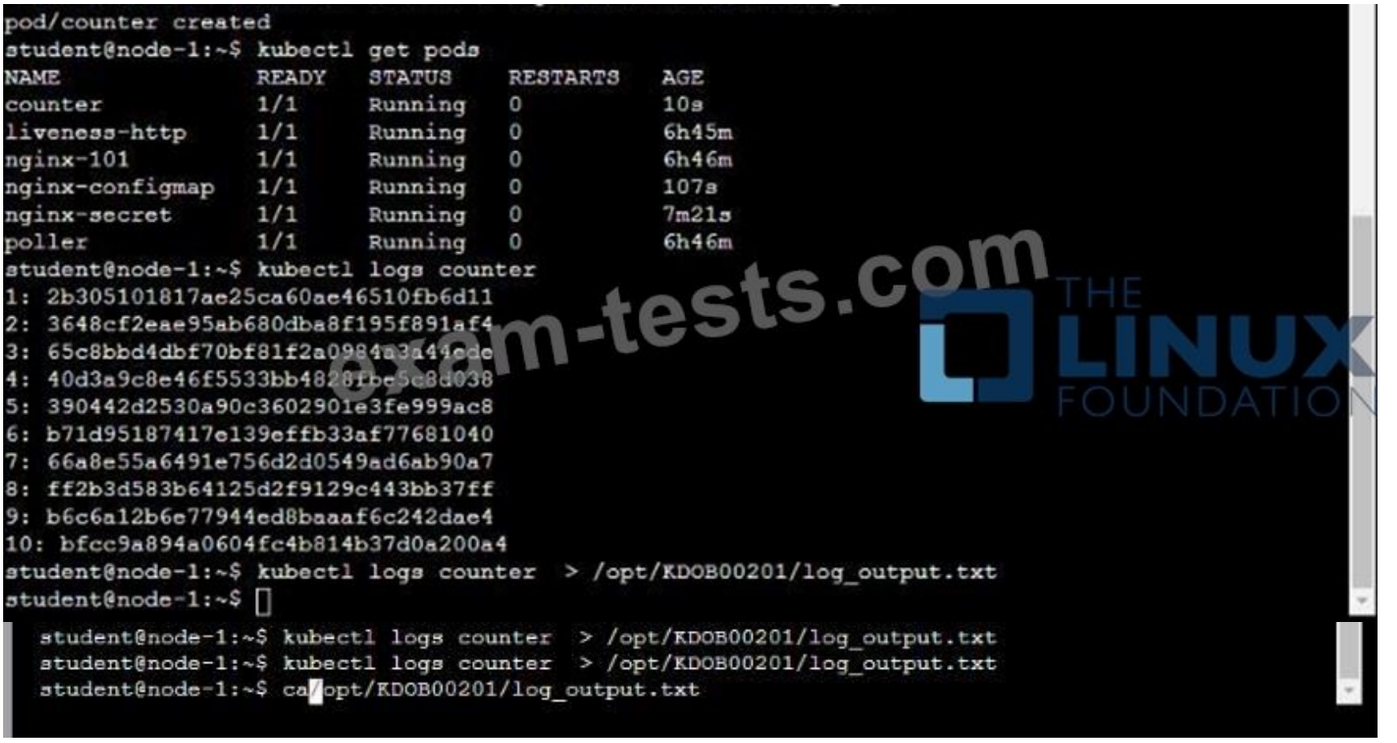
NEW QUESTION: 6

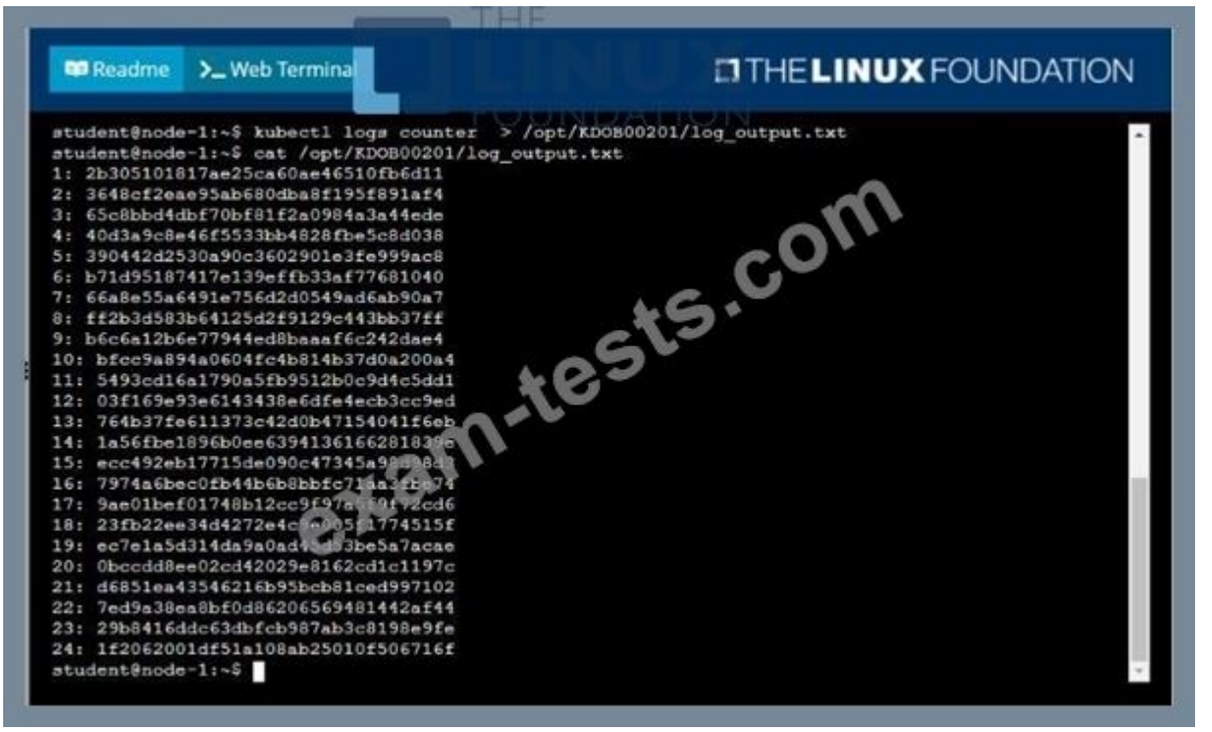
Context

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 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

Answer:

See the solution below.
Explanation
Solution:





NEW QUESTION: 7

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/KDOBG030l/pod.txt, which has already been created.

Answer:

See the solution below.

Explanation

Solution:

NEW QUESTION: 8

Refer to Exhibit.

Task:

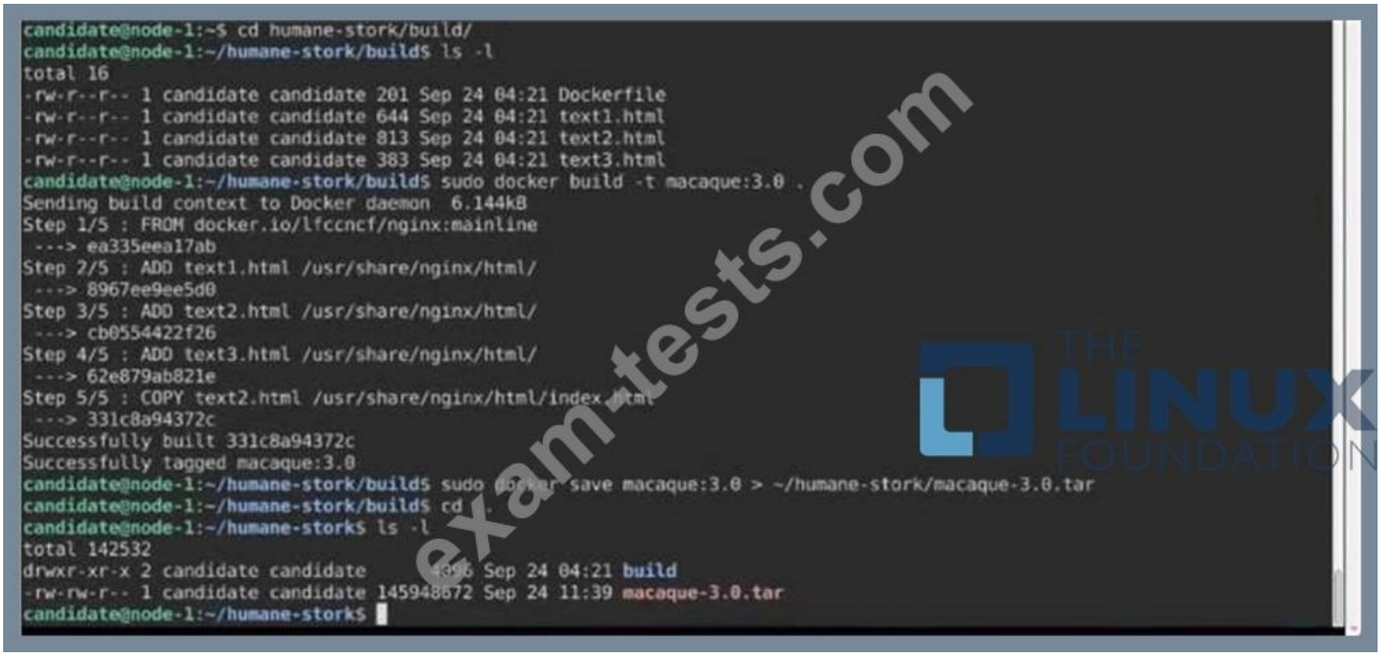
A Dockerfile has been prepared at -/human-stork/build/Dockerfile

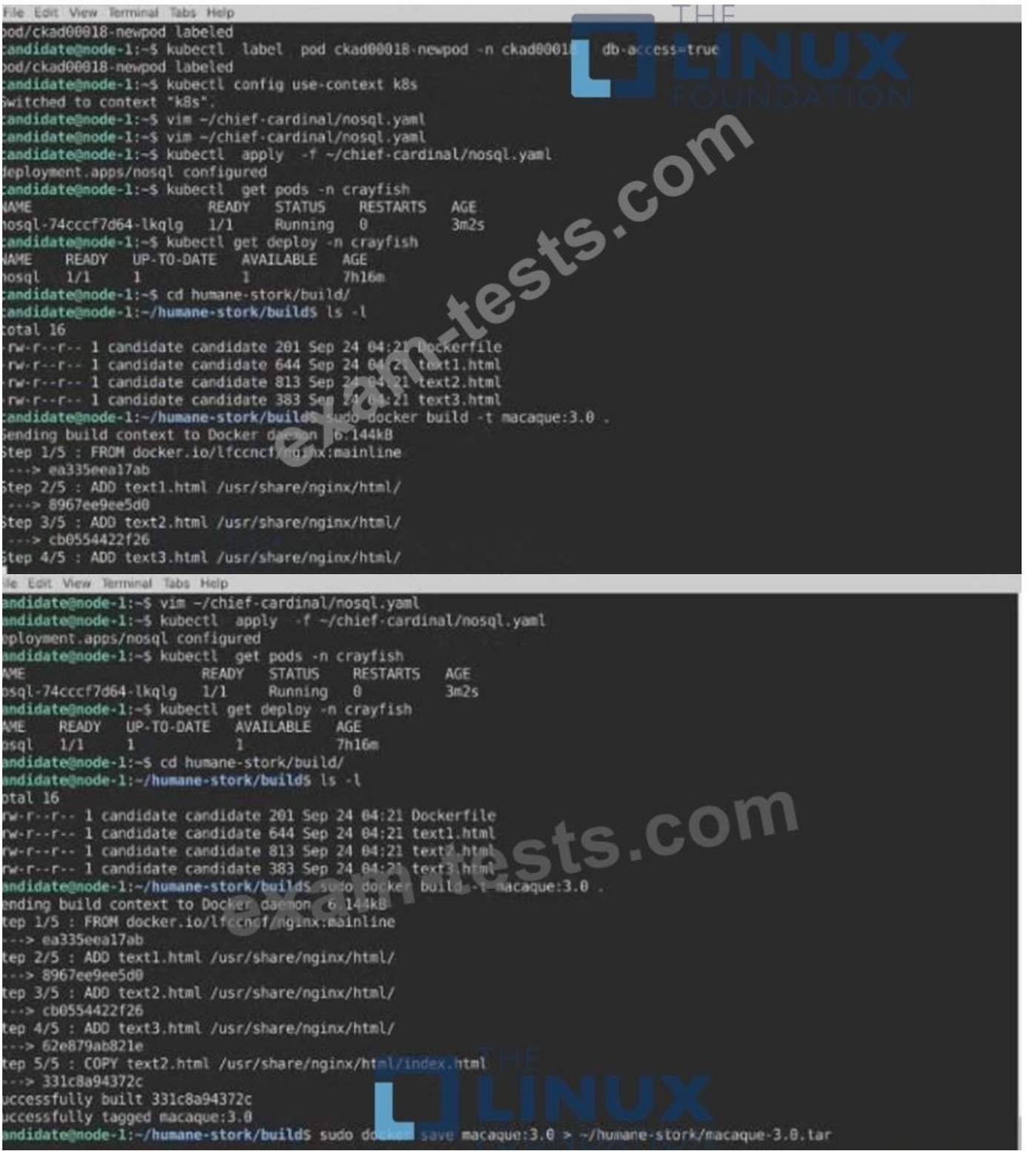
1) Using the prepared Dockerfile, build a container image with the name macque and lag 3.0. You may install and use the tool of your choice.

2) Using the tool of your choice export the built container image in OC-format and store it at -/human stork/macque 3.0 tar

Answer:

Solution:





NEW QUESTION: 9

Context

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=8080 and also expose that port for the container above

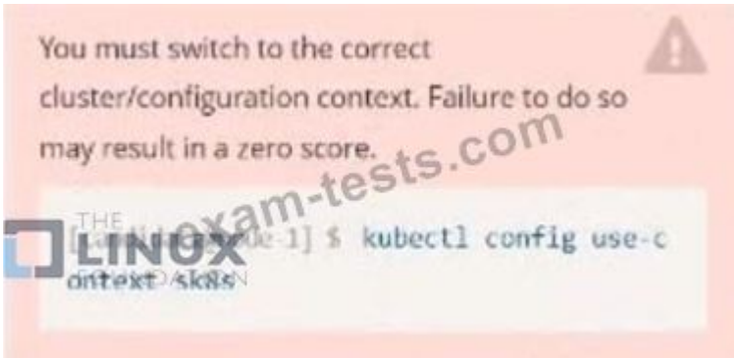
Answer:

Solution:


Readme
Web Terminal
THE LINUX FOUNDATION
apiVersion: apps/v1
kind: Deployment
metadata:
creationTimestamp: null
labels:
app: api
name: api
namespace: kdpd00201
spec:
replicas: 4
selector:
matchLabels:
app: api
strategy: {}
template:
metadata:
creationTimestamp: null
labels:
app: api
spec:
containers:
- image: lfccncf/nginx:1.13.7-alpine
name: nginx
resources: {}
status: {}
~
"nginx_deployment.yml" 25L, 421C
4,1
All
exam-tests.com
THE LINUX FOUNDATION


Readme
Web Terminal
THE LINUX FOUNDATION
apiVersion: apps/v1
kind: Deployment
metadata:
labels:
app: api
name: api
namespace: kdpd00201
spec:
replicas: 4
selector:
matchLabels:
app: api
template:
metadata:
labels:
app: api
spec:
containers:
- image: lfccncf/nginx:1.13.7-alpine
name: nginx
ports:
- containerPort: 8080
env:
- name: NGINX_PORT
value: "8080"
~
23,8
All
exam-tests.com
LINUX FOUNDATION

NEW QUESTION: 10

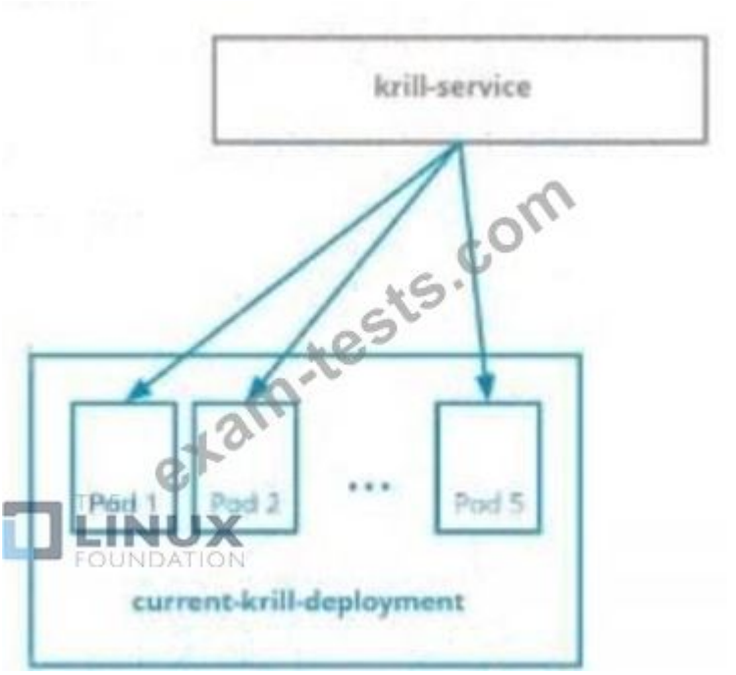


Context

You are asked to prepare a Canary deployment for testing a new application release.

Task:

A Service named krill-Service in the goshark namespace points to 5 pod created by the Deployment named current-krill-deployment

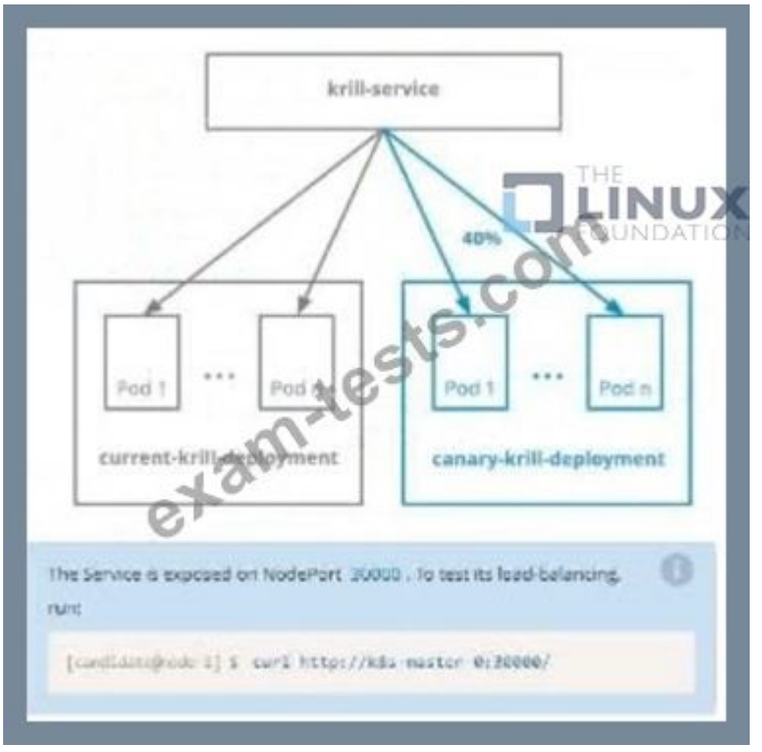


1) Create an identical Deployment named canary-kill-deployment, in the same namespace.

2) Modify the Deployment so that:

-A maximum number of 10 pods run in the goshawk namespace.

-40% of the krill-service 's traffic goes to the canary-krill-deployment pod(s)



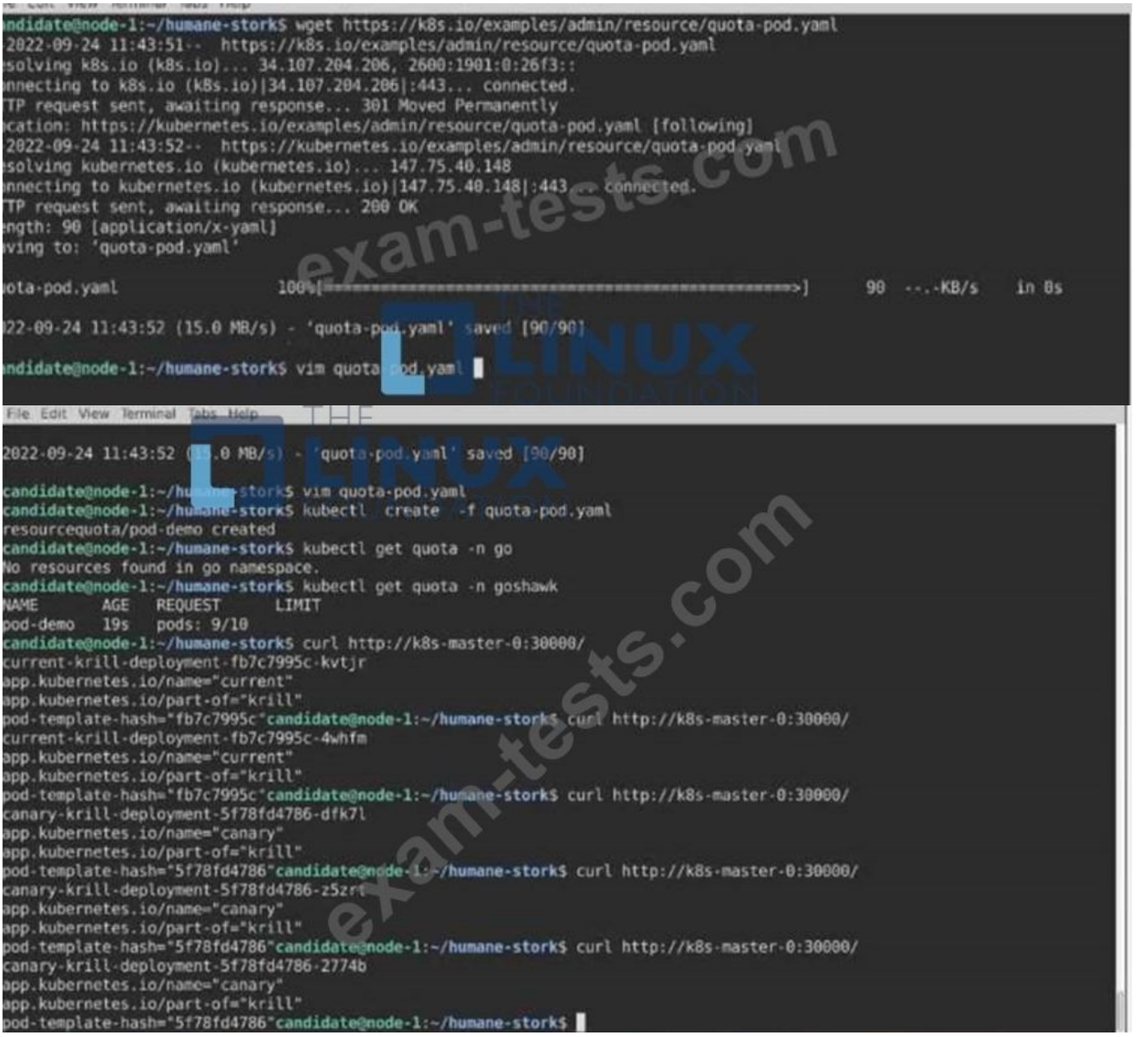
Answer:

See the solution below.

Explanation

Solution:

Text Description automatically generated



NEW QUESTION: 11

Context

Context

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

Task

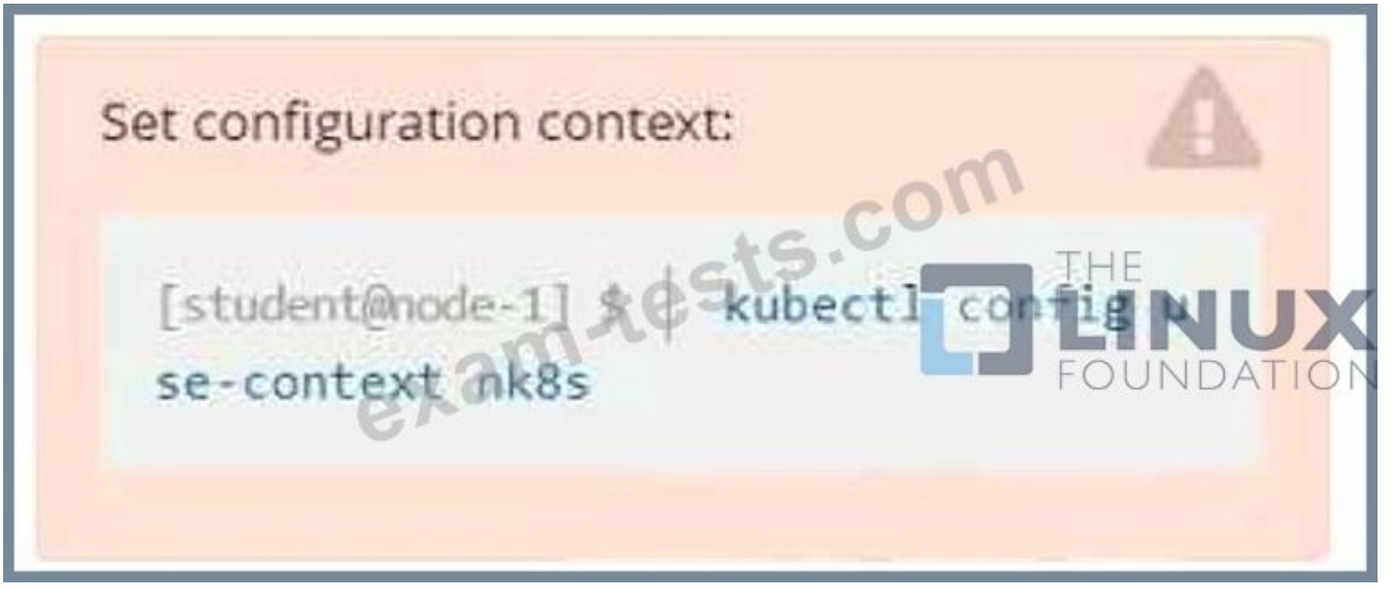
* From the pods running in namespace cpu-stress , write the name only of the pod that is consuming the most CPU to file /opt/KDOBG030l/pod.txt, which has already been created.

Answer:

Solution:

NEW QUESTION: 12

Context



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

Answer:

create deploy hello-deploy --image=nginx --dry-run=client -o yaml > hello-deploy.yaml Update deployment image to nginx:1.17.4: kubectl set image deploy/hello-deploy nginx=nginx:1.17.4

NEW QUESTION: 13

Context
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 See the solution below.

Answer:

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 -l run=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 -l run=my-nginx -o yaml | grep podIP
podIP: 10.244.3.4
podIP: 10.244.2.5

NEW QUESTION: 14
Refer to Exhibit.

Task:
Modify the existing Deployment named broker-deployment running in namespace quetzal so that its containers.

1) Run with user ID 30000 and

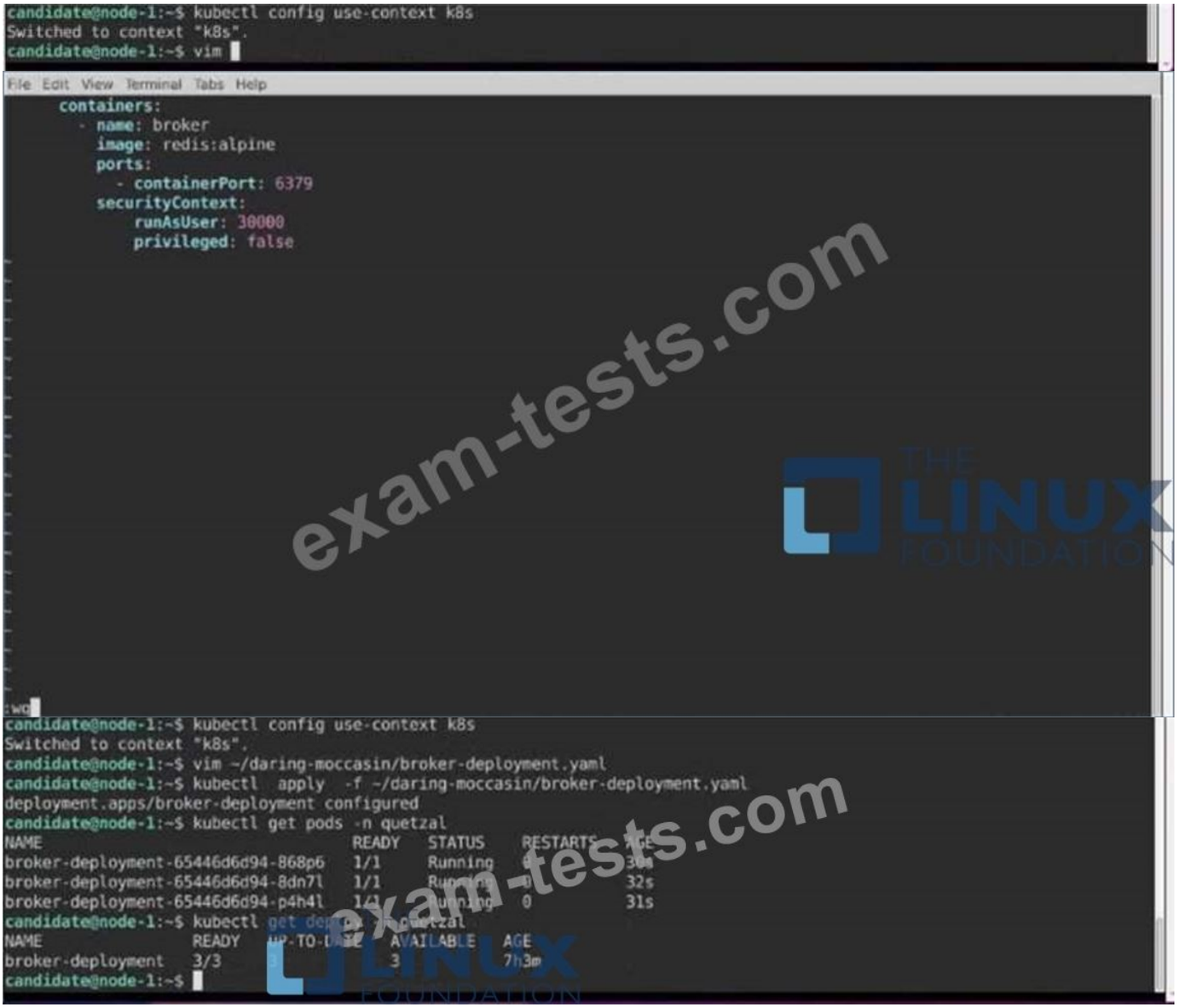
2) Privilege escalation is forbidden

The broker-deployment is manifest file can be found at:

~/daring-moccasin/broker-deployment.yaml

Answer:

Solution:



NEW QUESTION: 15

Exhibit:

Context

You sometimes need to observe a pod's logs, and write those logs to a file for further analysis.

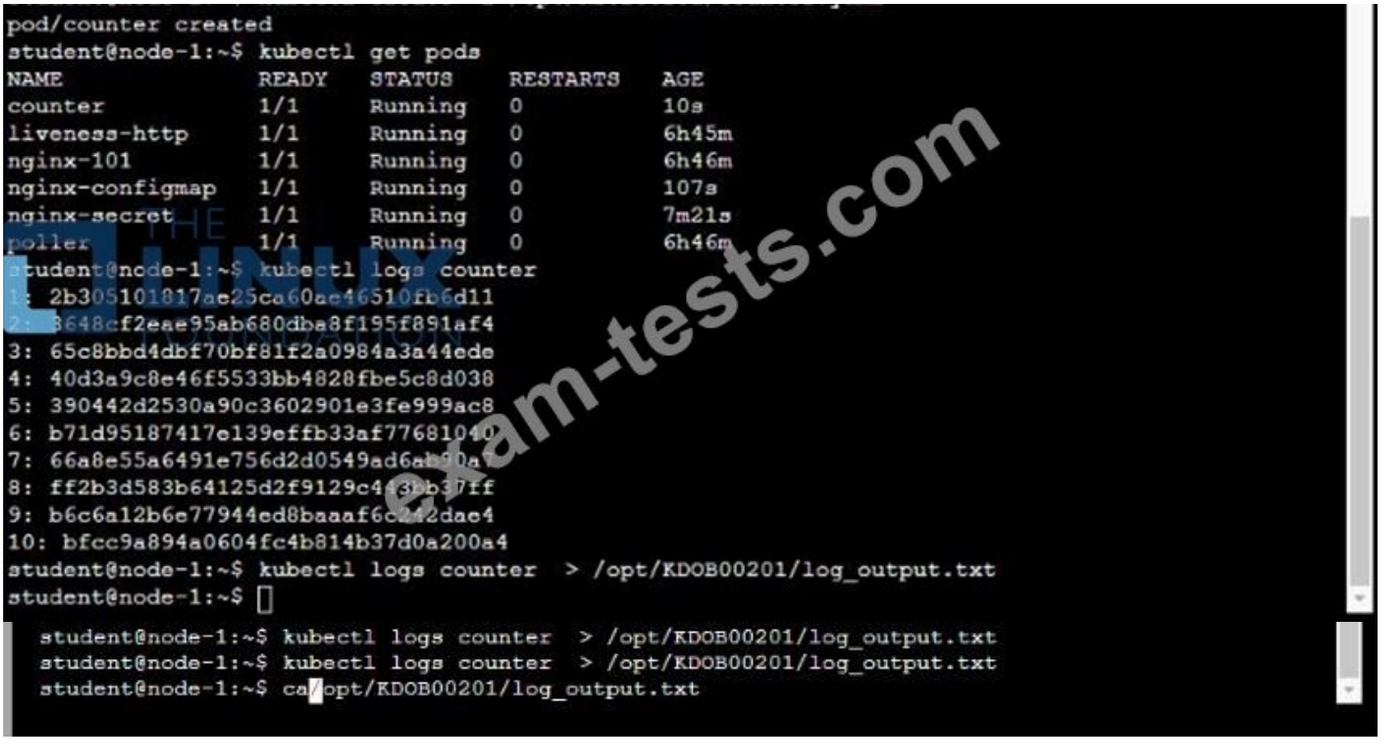
Task

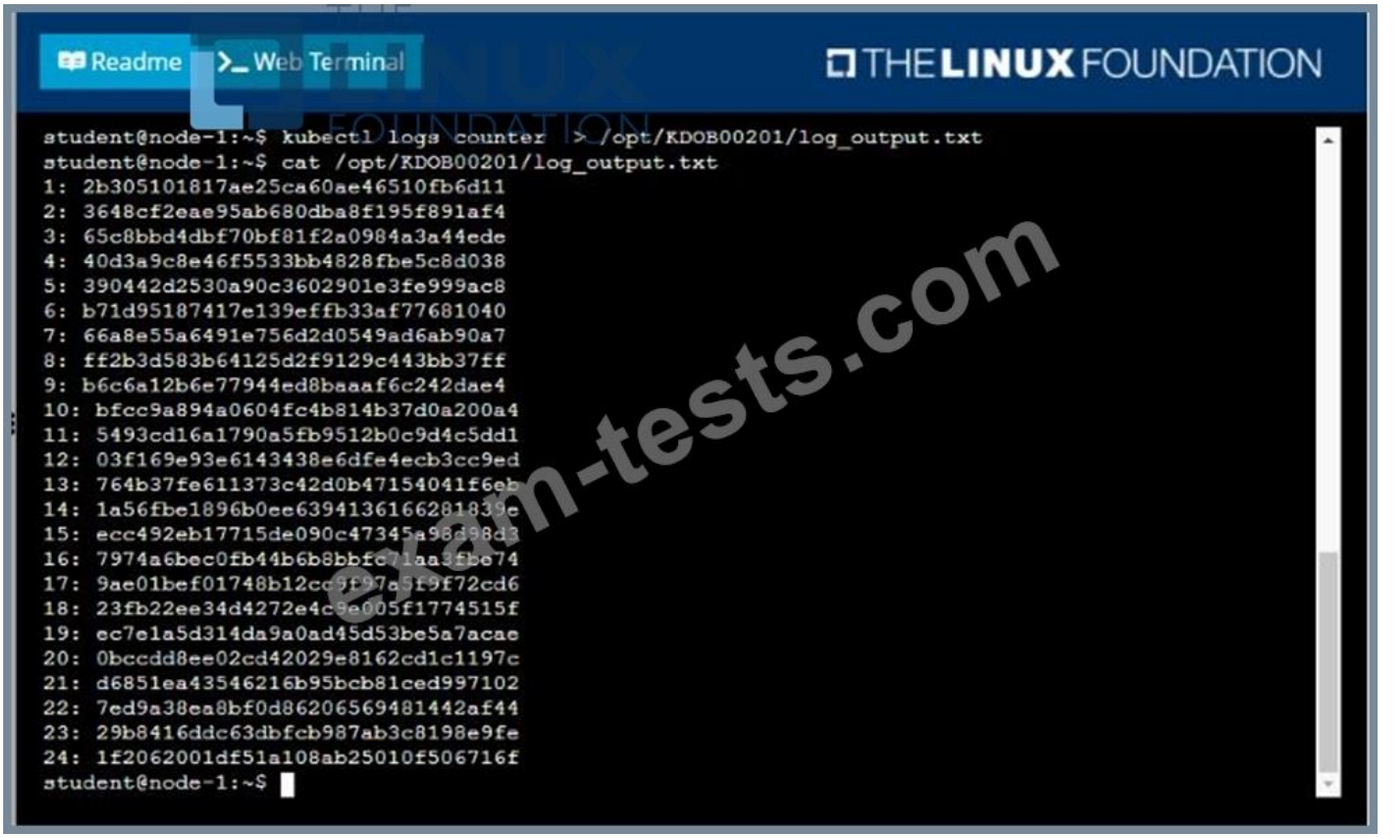
Please complete the following;

* Deploy the counter pod to the cluster using the provided 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

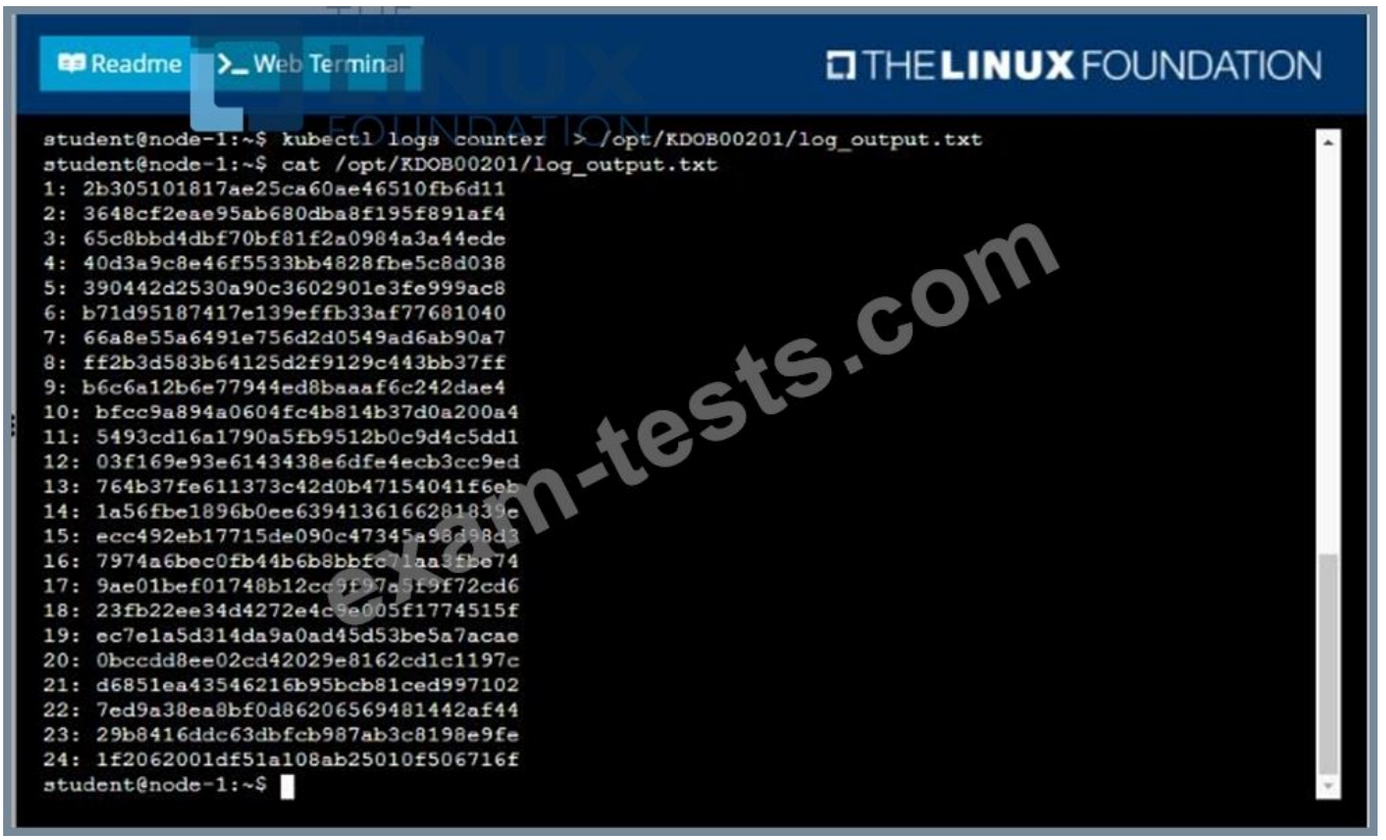
A. Solution:




Readme
Web Terminal
THE LINUX FOUNDATION
student@node-1:~$ kubectl logs counter > /opt/KDOB00201/log_output.txt
student@node-1:~$ cat /opt/KDOB00201/log_output.txt
1: 2b305101817ae25ca60ae46510fb6d11
2: 3648cf2eae95ab680dba8f195f891af4
3: 65c8bbd4dbf70bf81f2a0984a3a44ede
4: 40d3a9c8e46f5533bb4828fbe5c8d038
5: 390442d2530a90c3602901e3fe999ac8
6: b71d95187417e139effb33af77681040
7: 66a8e55a6491e756d2d0549ad6ab90a7
8: ff2b3d583b64125d2f9129c443bb37ff
9: b6c6a12b6e77944ed8baaaf6c242dae4
10: bfcc9a894a0604fc4b814b37d0a200a4
11: 5493cd16a1790a5fb9512b0c9d4c5dd1
12: 03f169e93e6143438e6dfe4ecb3cc9ed
13: 764b37fe611373c42d0b47154041f6eb
14: 1a56fbe1896b0ee63941361662818 39e
15: ecc492eb17715de090c47345a98d98d3
16: 7974a6bec0fb44b6b8bbfc71aa3fbe74
17: 9ae01bef01748b12cc9f97a5f9f72cd6
18: 23fb22ee34d4272e4c8e005f1774515f
19: ec7e1a5d314da9a0ad45d53be5a7acae
20: 0bccdd8ee02cd42029e8162cd1c1197c
21: d6851ea43546216b95bcb81ced997102
22: 7ed9a38ea8bf0d86206569481442af44
23: 29b8416ddc63dbfcb987ab3c8198e9fe
24: 1f2062001df51a108ab25010f506716f
student@node-1:~$

B. Solution:


pod/counter created
student@node-1:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
counter 1/1 Running 0 10s
liveness-http 1/1 Running 0 6h45m
nginx-101 1/1 Running 0 6h46m
nginx-configmap 1/1 Running 0 107s
nginx-secret 1/1 Running 0 7m21s
poller 1/1 Running 0 6h46m
student@node-1:~$ kubectl logs counter
1: 2b305101817ae25ca60ae46510fb6d11
2: 3648cf2eae95ab680dba8f195f891af4
3: 65c8bbd4dbf70bf81f2a0984a3a44ede
4: 40d3a9c8e46f5533bb4828fbe5c8d038
5: 390442d2530a90c3602901e3fe999ac8
6: b71d95187417e139effb33af77681040
7: 66a8e55a6491e756d2d0549ad6ab90a7
8: ff2b3d583b64125d2f9129c443bb37ff
9: b6c6a12b6e77944ed8baaaf6c242dae4
10: bfcc9a894a0604fc4b814b37d0a200a4
student@node-1:~$ kubectl logs counter > /opt/KDOB00201/log_output.txt
student@node-1:~$



Answer: A (LEAVE A REPLY)

NEW QUESTION: 16

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 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

A. Solution:

B. Solution:

Answer: A (LEAVE A REPLY)

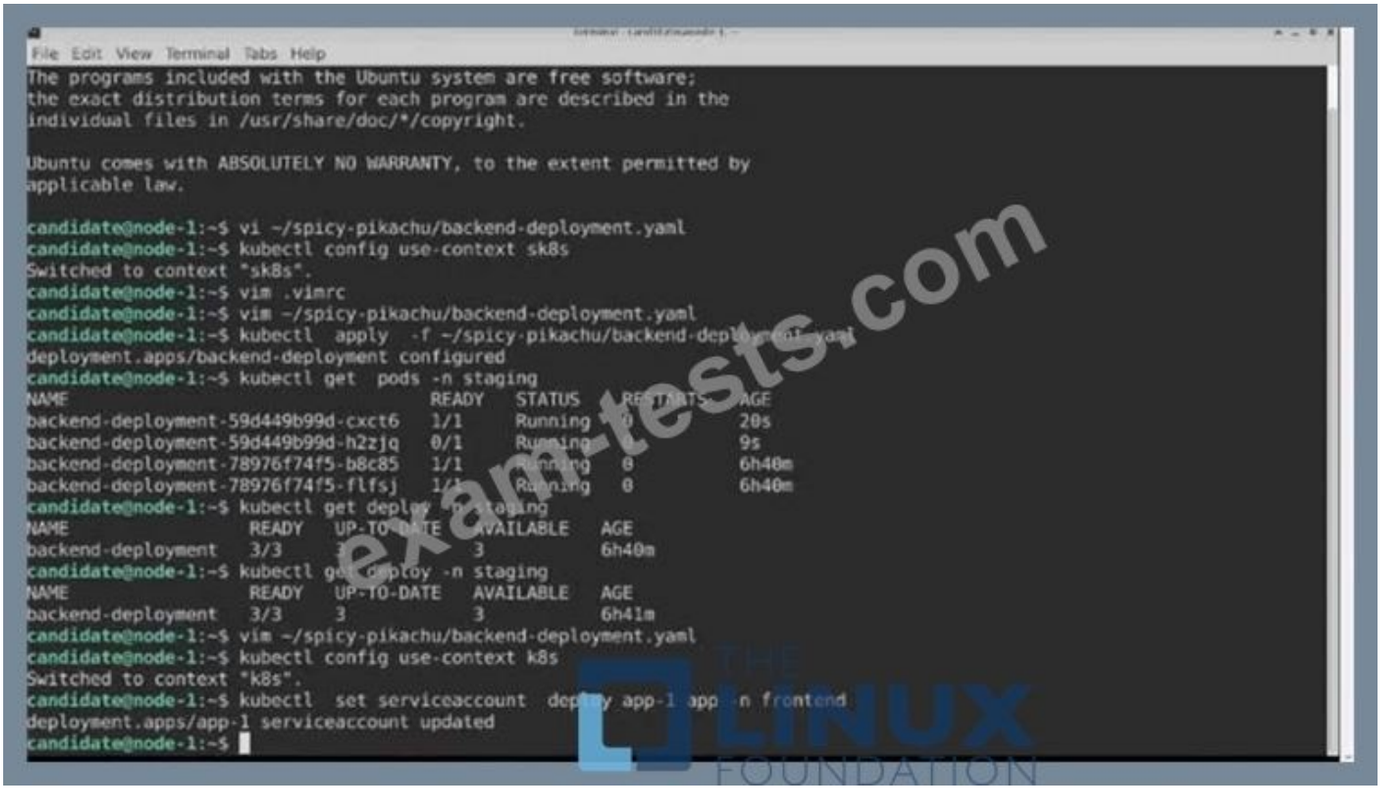
NEW QUESTION: 17
Refer to Exhibit.

Task:

Update the Deployment app-1 in the frontend namespace to use the existing ServiceAccount app.

Answer:

Solution:



NEW QUESTION: 18

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:

```
<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.

Answer:

See the solution below.

Explanation

Solution:

Create the Pod:

kubectl create

-f http://k8s.io/docs/tasks/configure-pod-container/

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 "gcr.io/google_containers/busybox"

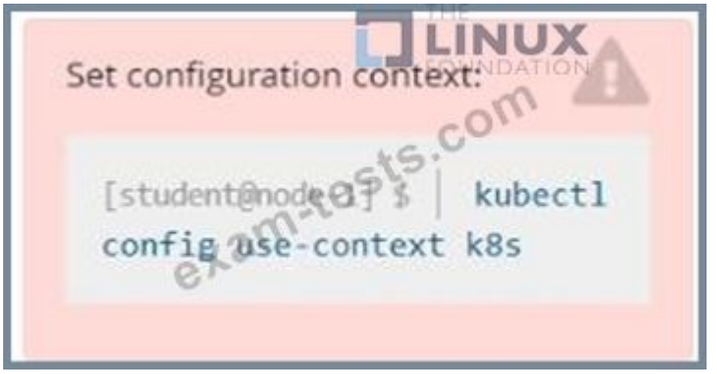
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image "gcr.io/google_containers/busybox"

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 "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 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 m

NEW QUESTION: 19
Refer to 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 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
```

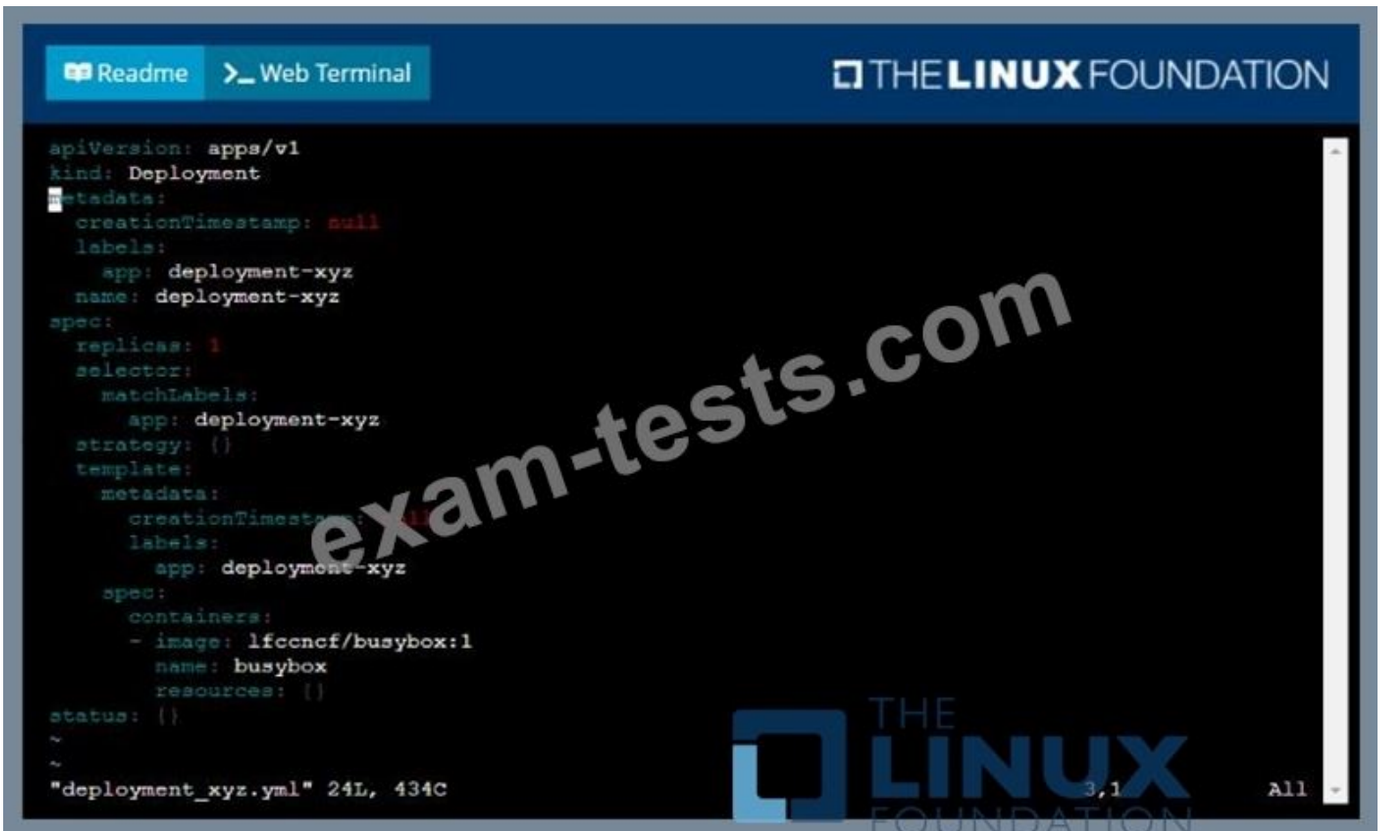
which should output logs to /tmp/log/input.log in plain text format, with example values:

```
i luv cncf
i luv cncf
i luv cncf
```

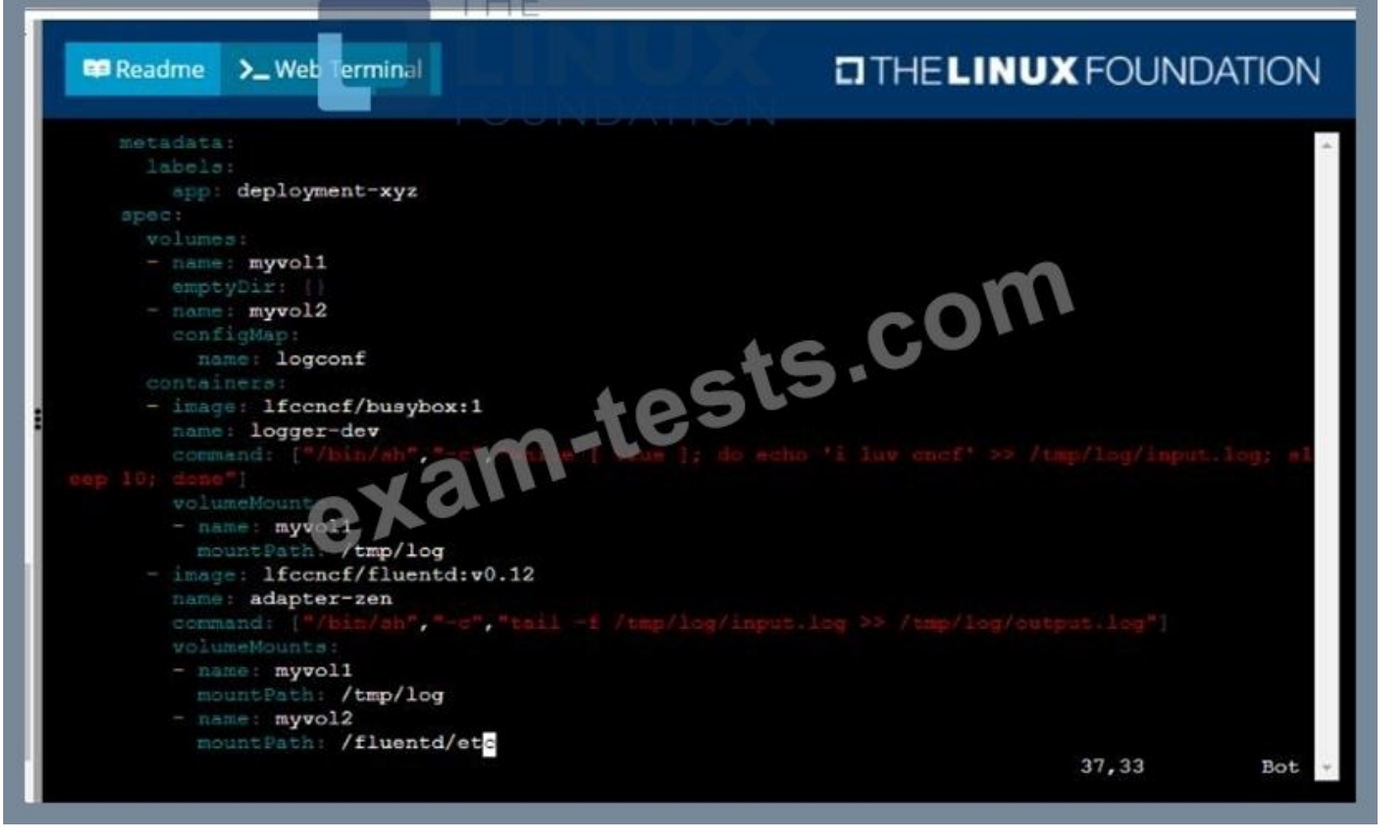
* The adapter-zen sidecar container should read /tmp/log/input.log and output the data to /tmp/log/output.* in Fluentd JSON format. Note that no knowledge of Fluentd is required to complete this task: all you will need to achieve this is to create the ConfigMap from the spec file provided at /opt/KDMC00102/fluentd-configma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

Answer:

Solution:


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: lfccncf/busybox:1
name: busybox
resources: {}
status: {}
~
~
"deployment_xyz.yml" 24L, 434C
3,1
All
exam-tests.com
THE LINUX FOUNDATION


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
exam-tests.com
THE LINUX FOUNDATION



NEW QUESTION: 20

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 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 See the solution below.

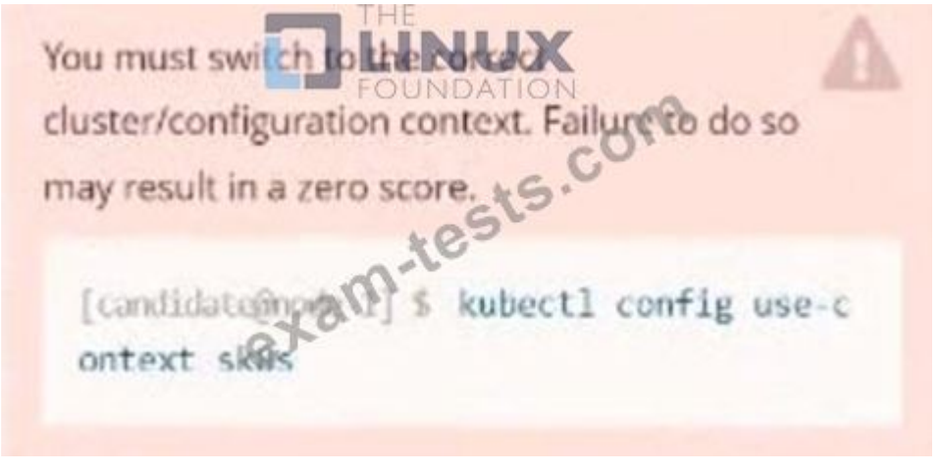
Answer:

Explanation

Solution:

NEW QUESTION: 21

Context

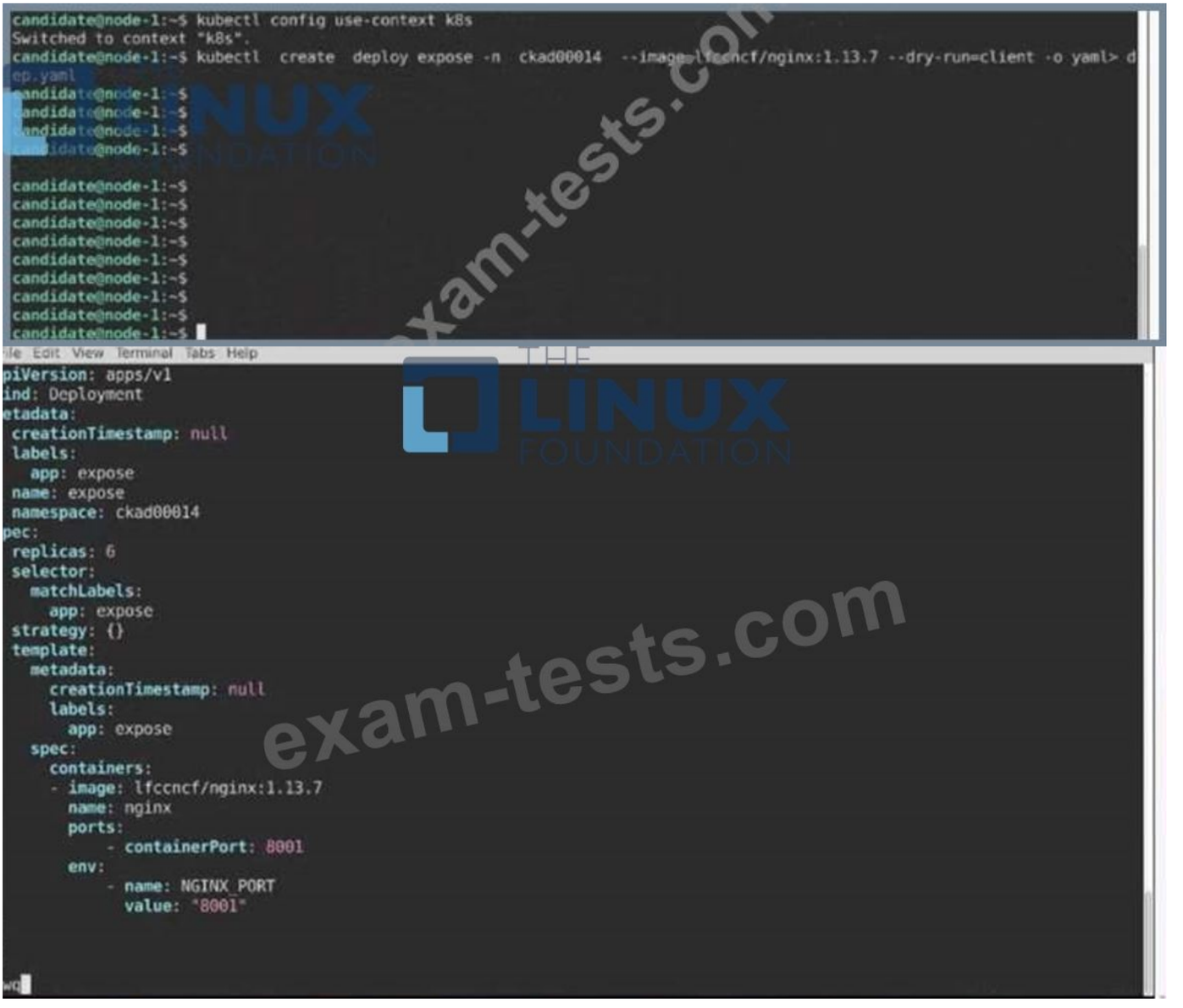


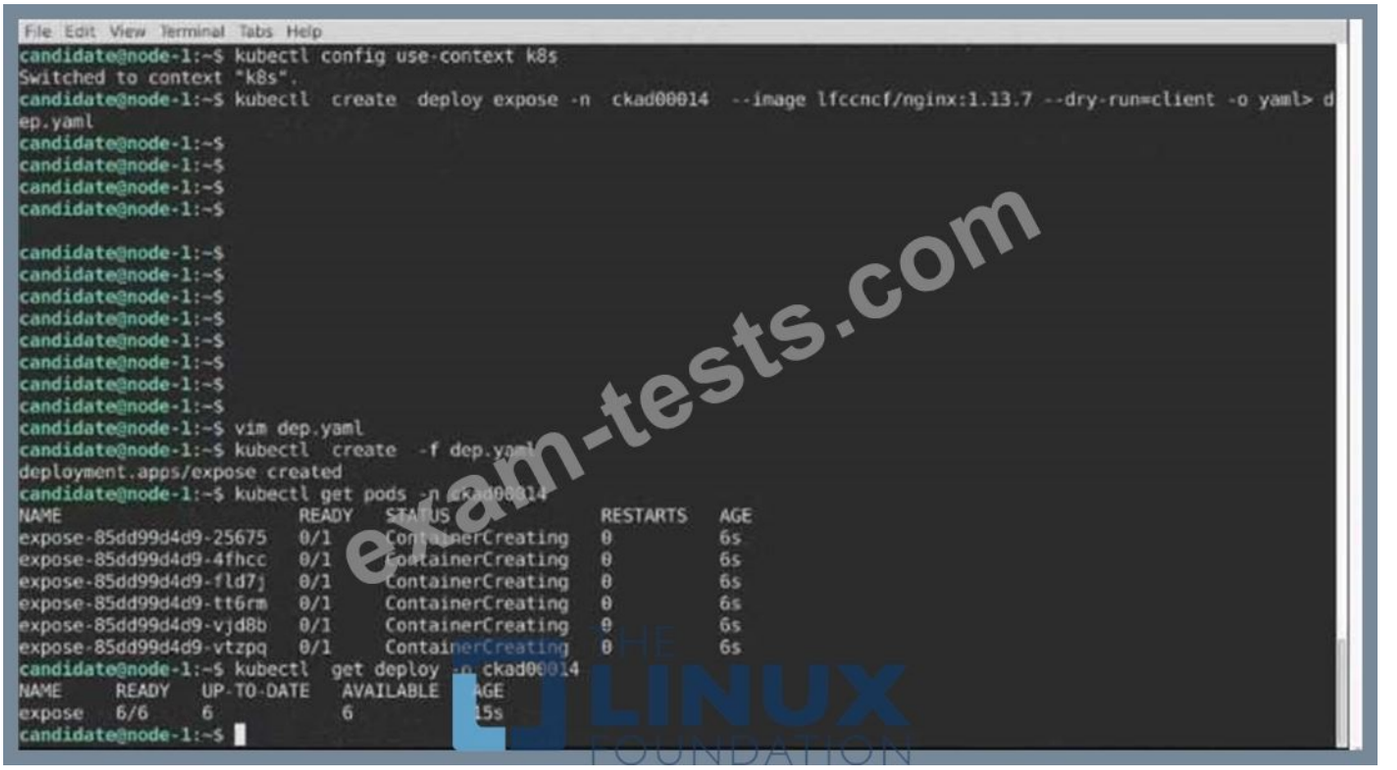
Task:

Create a Deployment named expose in the existing ckad00014 namespace running 6 replicas of a Pod. Specify a single container using the ifccncf/nginx: 1.13.7 image Add an environment variable named NGINX_PORT with the value 8001 to the container then expose port 8001

Answer:

Solution:





NEW QUESTION: 22

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. Solution:

```
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
```

Readme | Web Terminal — THE LINUX FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-secret
  name: nginx-secret
spec:
  containers:
  - image: nginx
    name: nginx-secret
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
~
~
~
~
~
~
~
~
~
~
~
"nginx_secret.yml" 15L, 253C                                  1,1           All
```

Readme | Web Terminal — THE LINUX FOUNDATION

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx-secret
  name: nginx-secret
spec:
  containers:
  - image: nginx
    name: nginx-secret
    env:
    - name: COOL_VARIABLE
      valueFrom:
        secretKeyRef:
          name: some-secret
          key: key1
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                16,20         All
```

B. Solution:

```
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
```

Answer: B (LEAVE A REPLY)

NEW QUESTION: 23

Exhibit:

Task

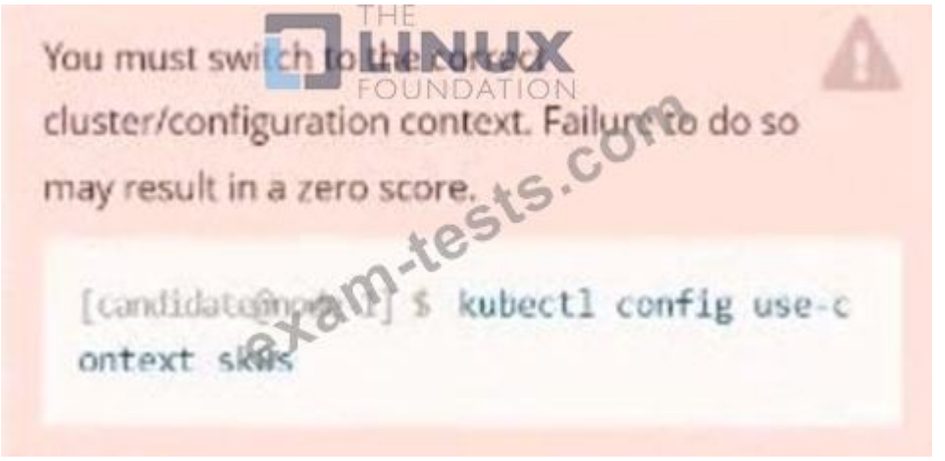
You have rolled out a new pod to your infrastructure and now you need to allow it to communicate with the web and storage pods but nothing else. Given the running pod kdsn00201 -newpod edit it to use a network policy that will allow it to send and receive traffic only to and from the web and storage pods.

A. Pending

Answer: (SHOW ANSWER)

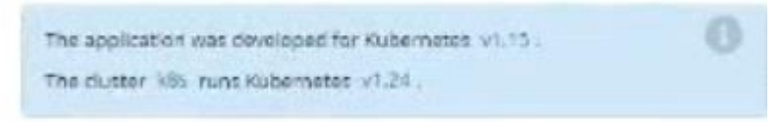
NEW QUESTION: 24

Context



Task:

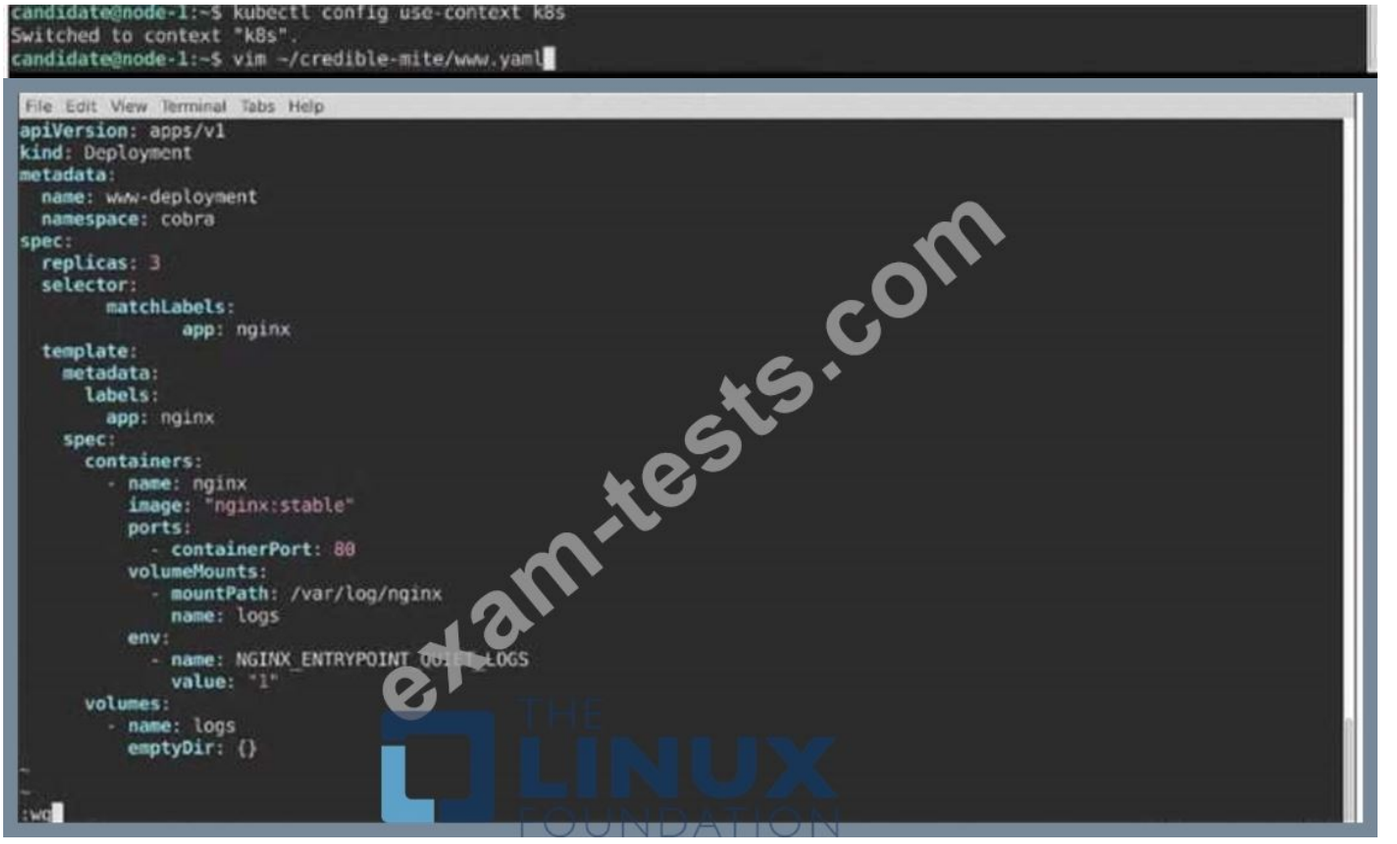
1) Fix any API depreciation issues in the manifest file -/credible-mite/www.yaml so that this application can be deployed on cluster K8s.

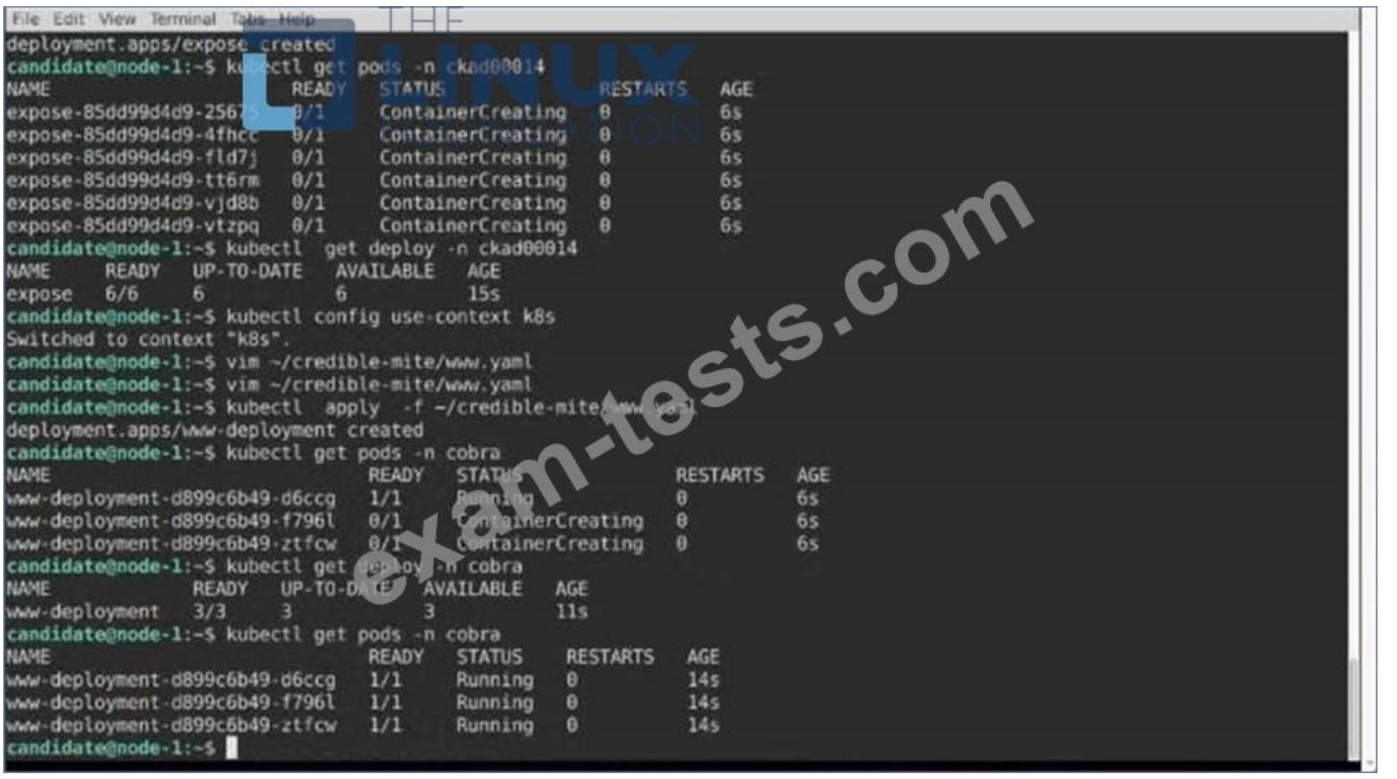


2) Deploy the application specified in the updated manifest file -/credible-mite/www.yaml in namespace cobra

Answer:

Solution:





NEW QUESTION: 25

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 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

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

A. Solution:

```
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:~$
```

B. Solution:

Readme | Web Terminal

THE LINUX FOUNDATION

```
student@node-1:~$ kubectl create deployment deployment-xyz --image=lfccncf/busybox:1 --dry-run=client -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: lfccncf/busybox:1
        name: busybox
        resources: {}
status: {}
~
~
"deployment_xyz.yml" 24L, 434C                                3,1           All
```

```
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:~$
```

Answer: A (LEAVE A REPLY)

NEW QUESTION: 26

Refer to 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

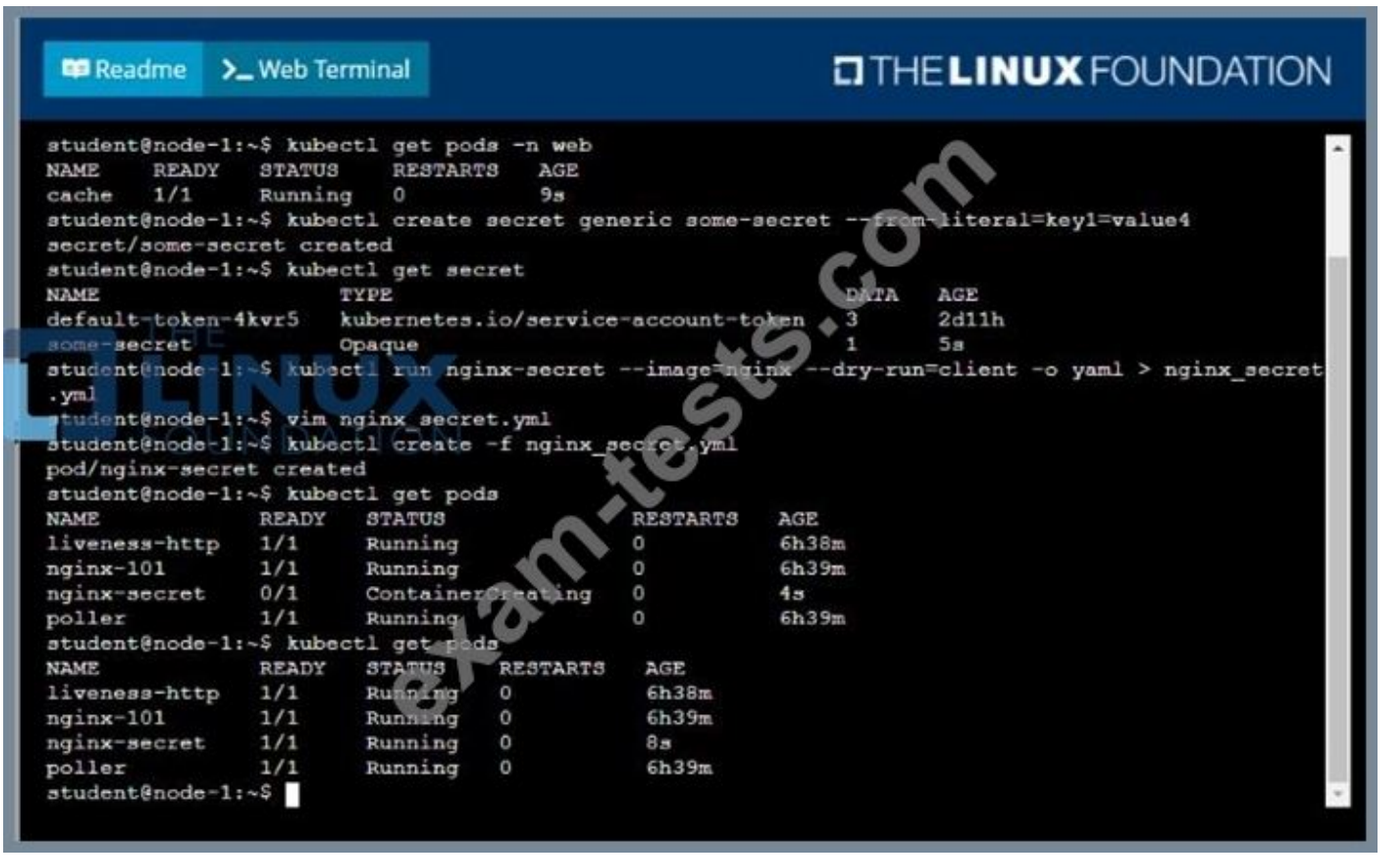
* Start an nginx pod named nginx-secret using container image nginx, and add an environment variable exposing the value of the secret key key 1, using COOL_VARIABLE as the name for the environment variable inside the pod

Answer:

Solution:


Readme
Web Terminal
THE LINUX FOUNDATION
apiVersion: v1
kind: Pod
metadata:
creationTimestamp: null
labels:
run: nginx-secret
name: nginx-secret
spec:
containers:
- image: nginx
name: nginx-secret
resources: {}
dnsPolicy: ClusterFirst
restartPolicy: Always
status: {}
"nginx_secret.yml" 15L, 253C
1,1
All
exam-tests.com


Readme
Web Terminal
THE LINUX FOUNDATION
apiVersion: v1
kind: Pod
metadata:
labels:
run: nginx-secret
name: nginx-secret
spec:
containers:
- image: nginx
name: nginx-secret
env:
- name: COOL_VARIABLE
valueFrom:
secretKeyRef:
name: some-secret
key: key1
-- INSERT --
16,20
All
exam-tests.com
THE LINUX FOUNDATION



NEW QUESTION: 27

Exhibit:

Context

You are tasked to create a ConfigMap and consume the ConfigMap in a pod using a volume mount.

Task

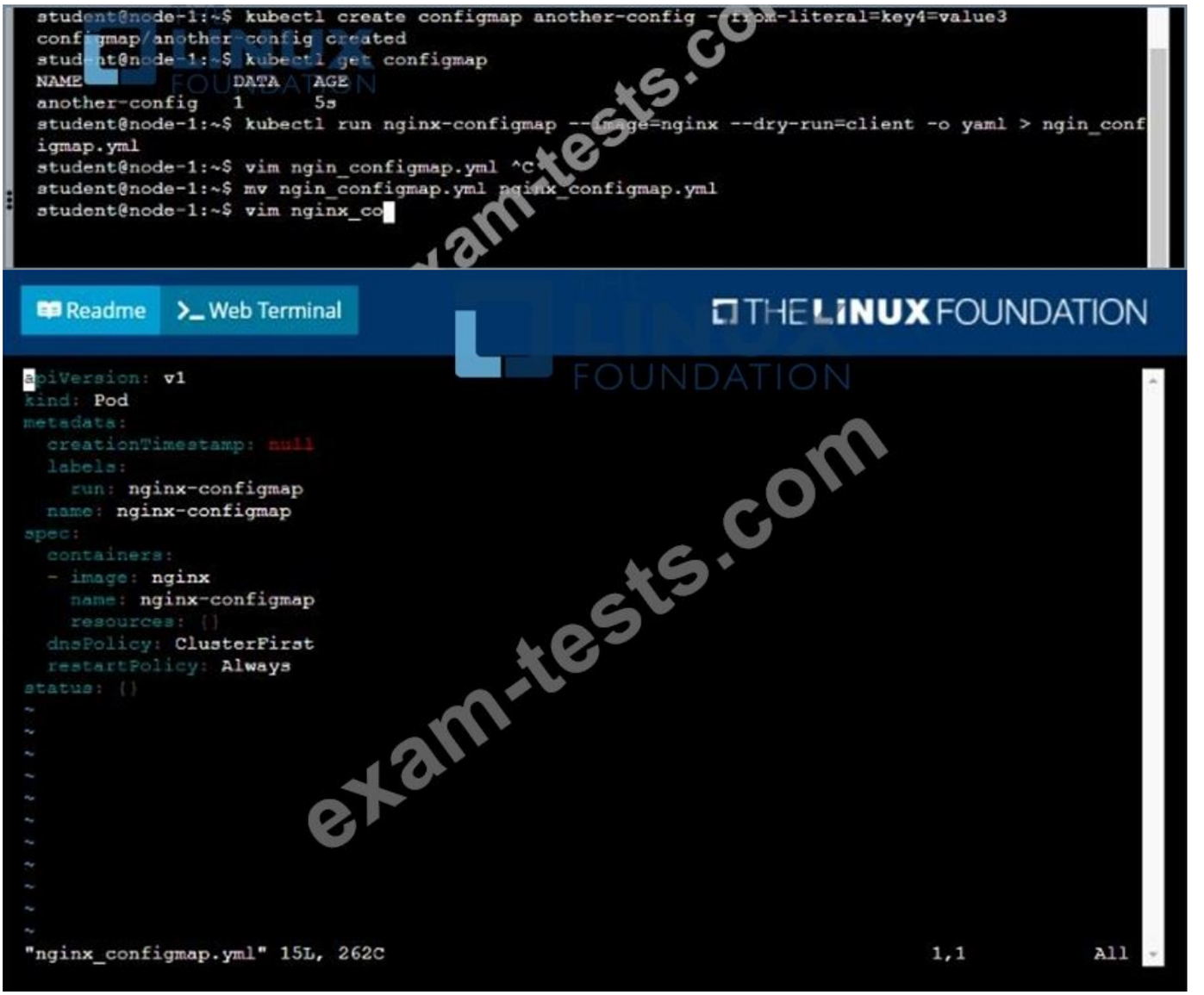
Please complete the following:

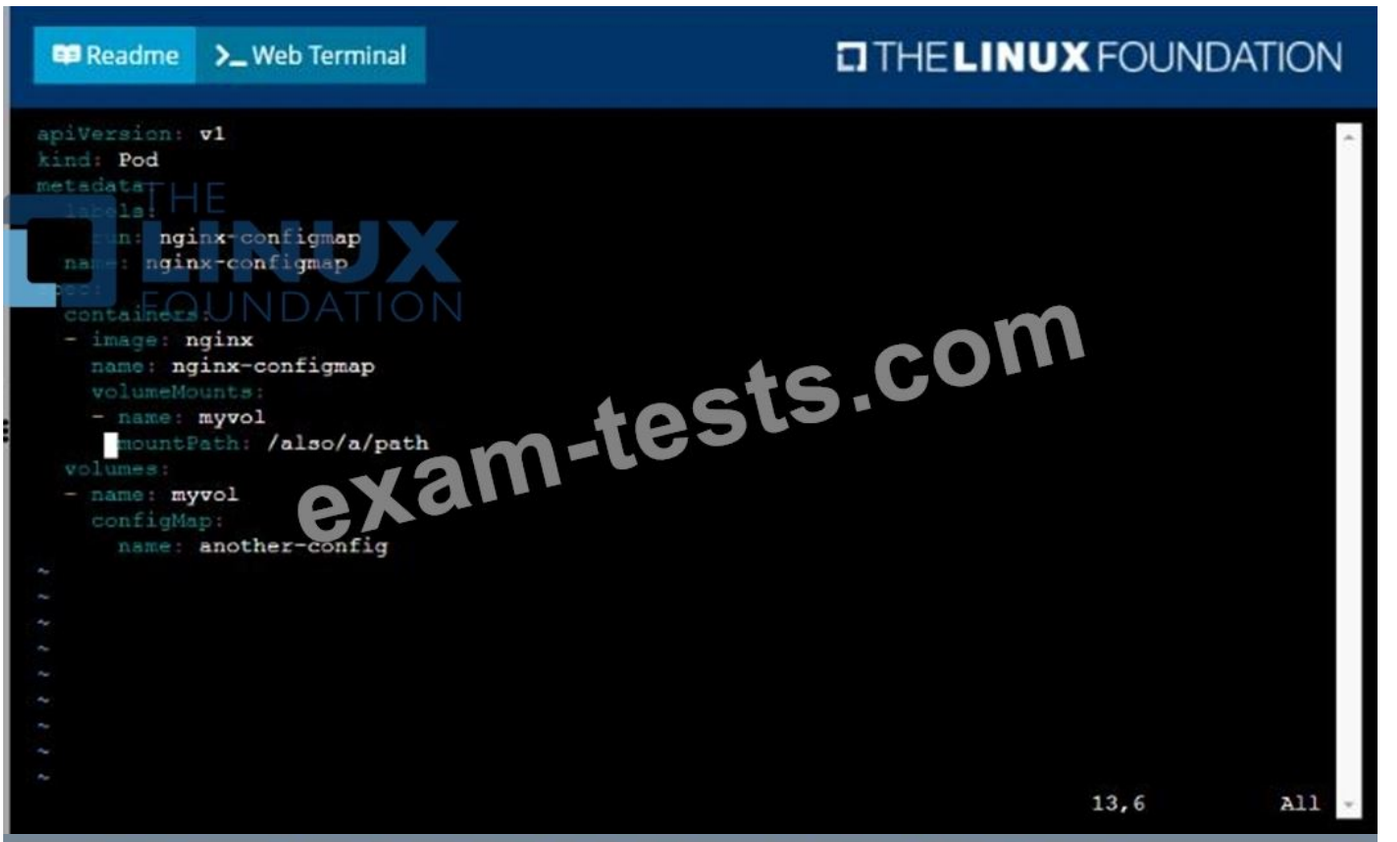
* Create a ConfigMap named another-config containing the key/value pair: key4/value3

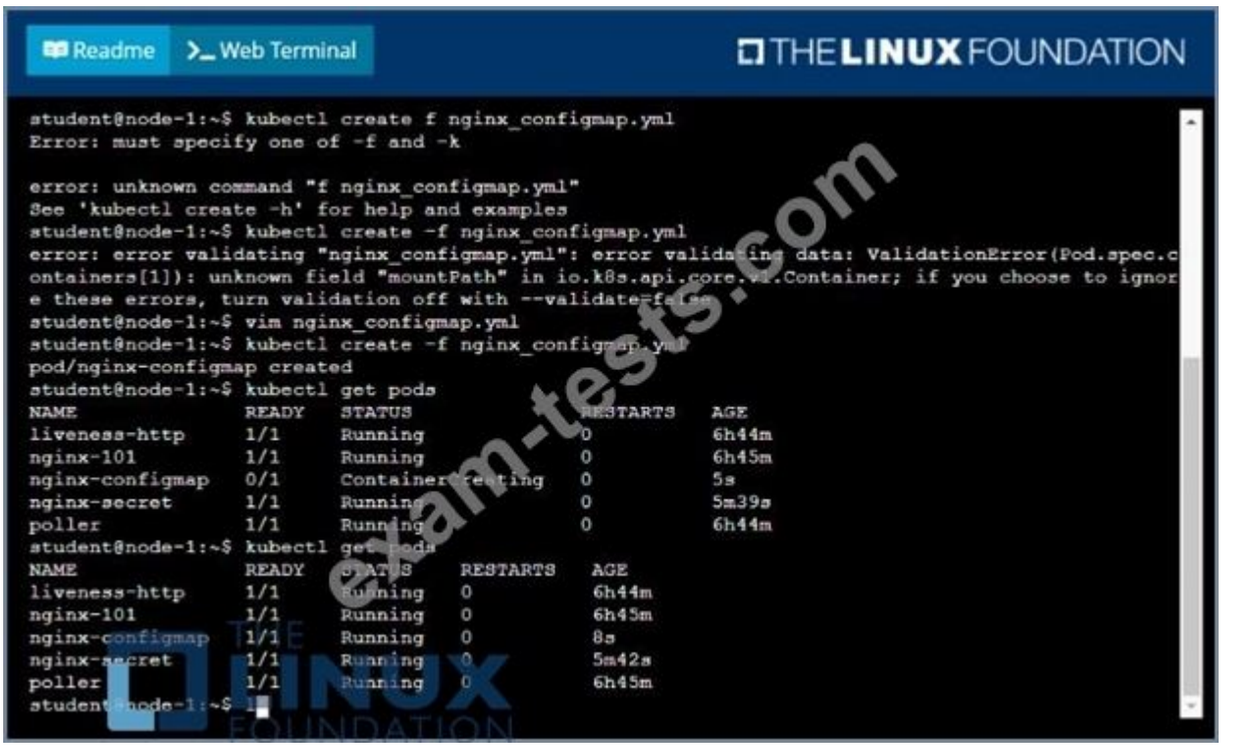
* start a pod named nginx-configmap containing a single container using the

nginx image, and mount the key you just created into the pod under directory /also/a/path

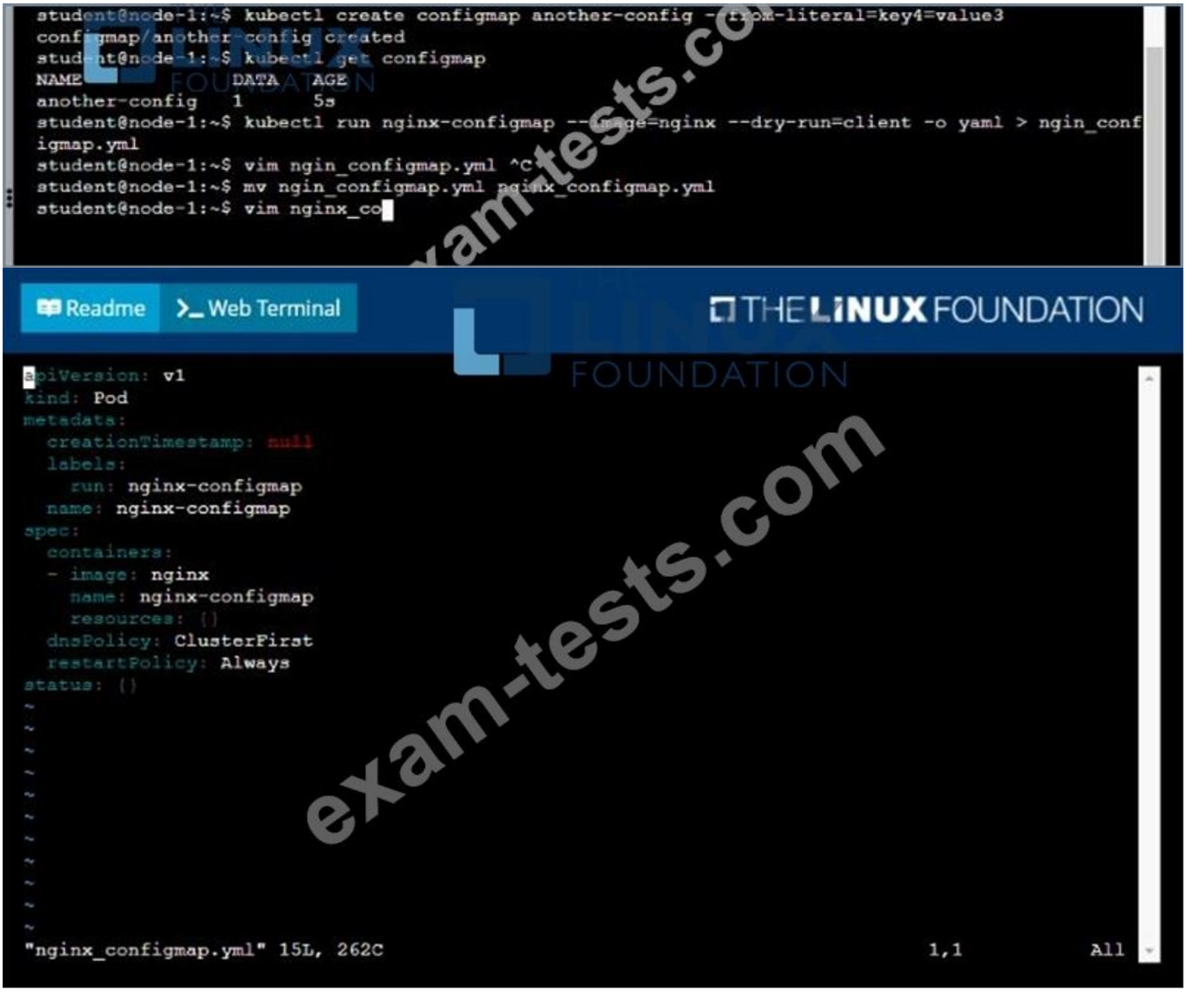
A. Solution:





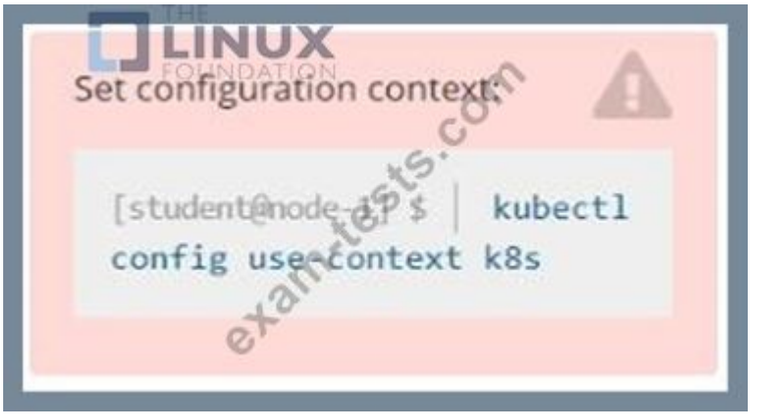


B. Solution:



Answer: (SHOW ANSWER)

NEW QUESTION: 28
Refer to Exhibit.



Set Configuration Context:
[student@node-1] $ | kubectl
Config use-context k8s
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 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 the 3.2 tag
* Expose port 6379
Answer:
Solution:

NEW QUESTION: 29
Refer to Exhibit.

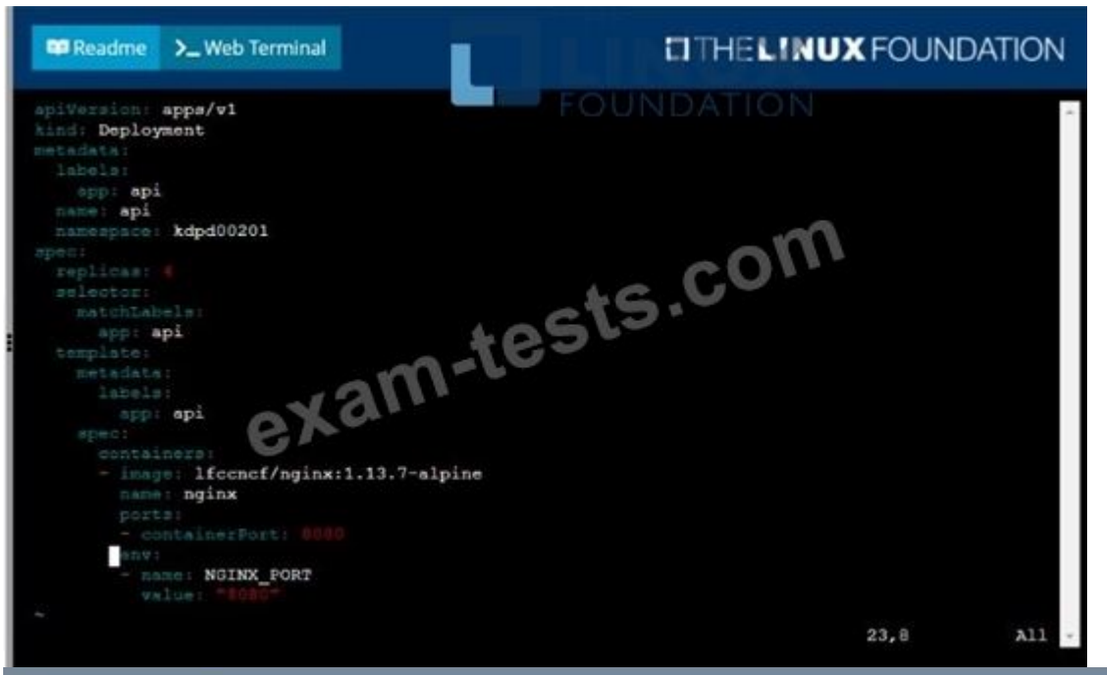
Task

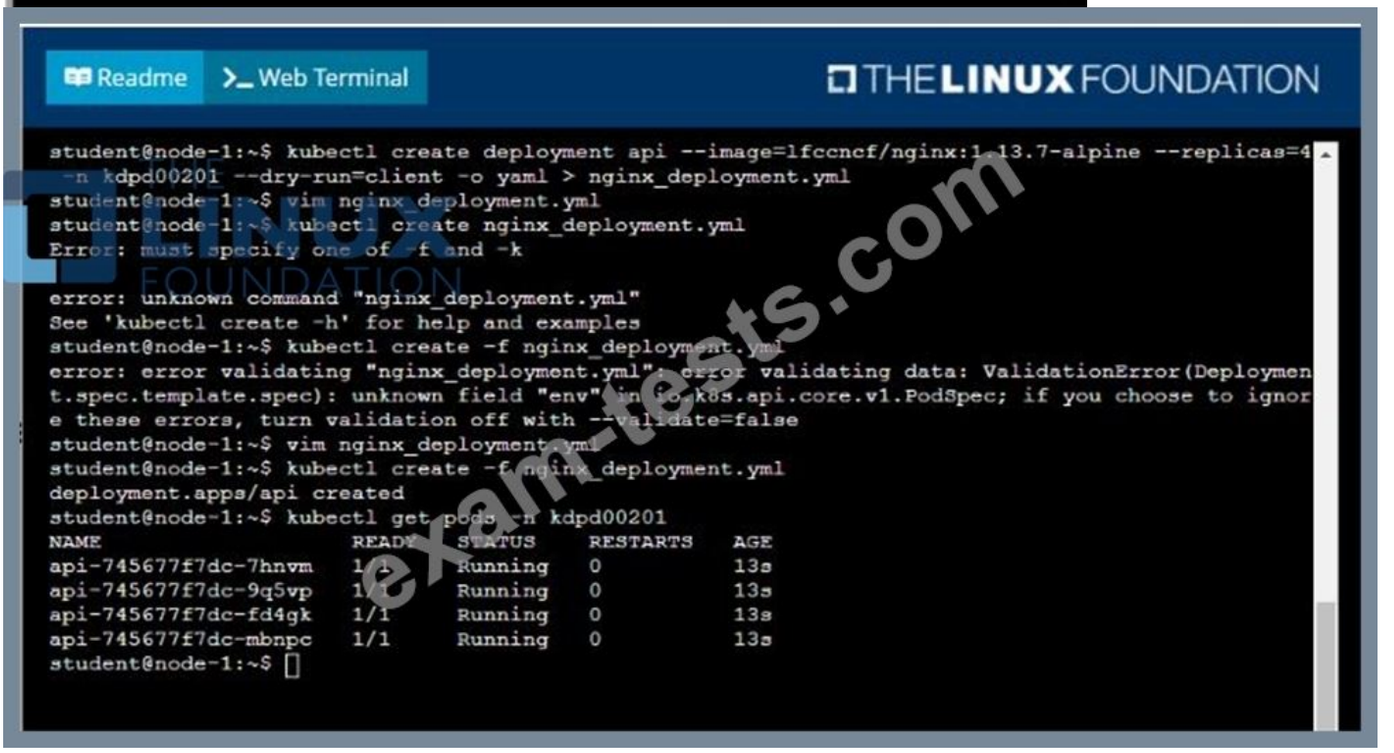
Create a new deployment for running.nginx with the following parameters;

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

Answer:

Solution:





NEW QUESTION: 30

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:

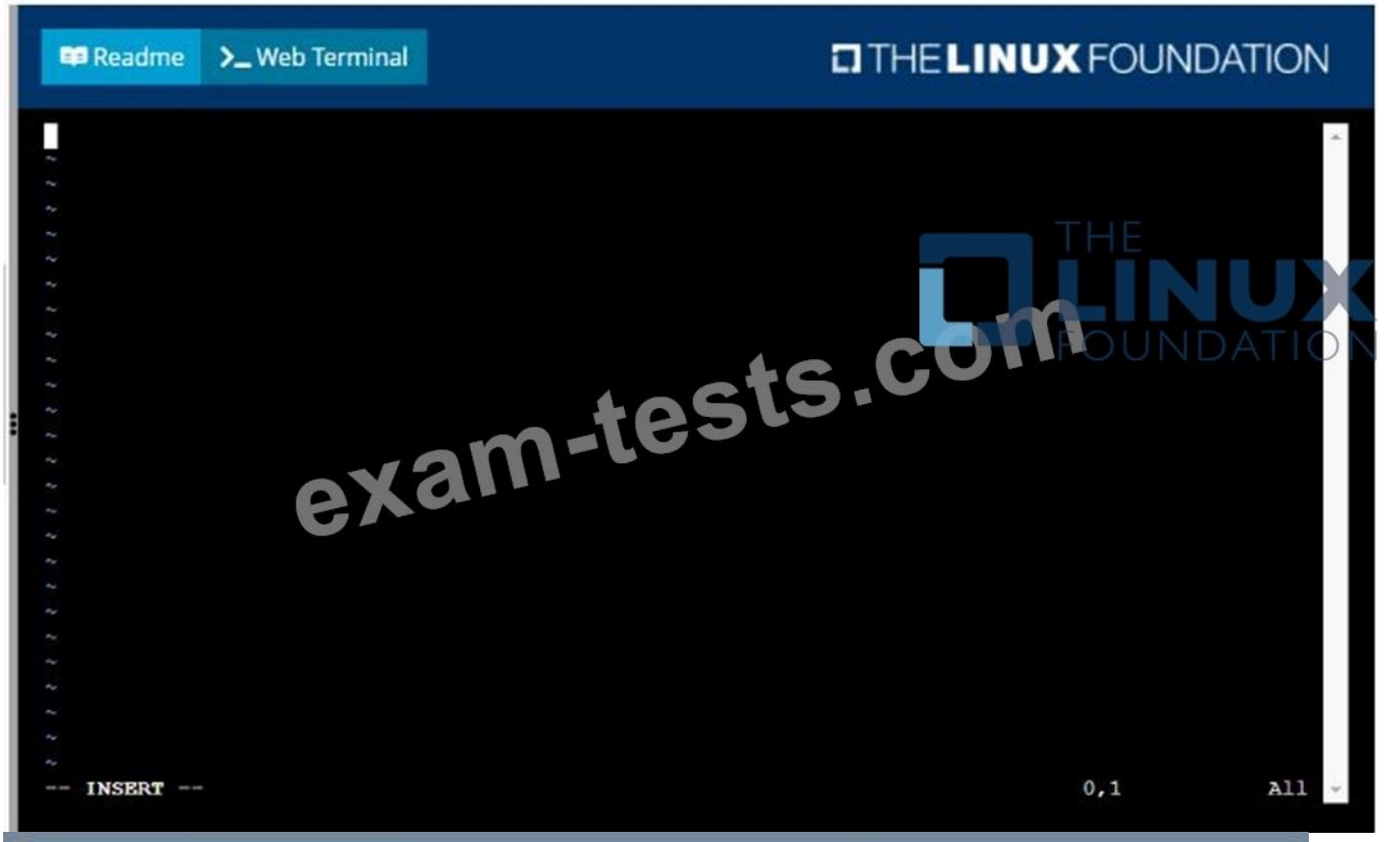
* Create a file on node sk8s-node-0 at /opt/KDSP00101/data/index.html with the content Acct=Finance
* Create a PersistentVolume named task-pv-volume using hostPath and allocate 1Gi to it, specifying that the volume is at /opt/KDSP00101/data on the cluster's node. The configuration should specify the access mode of ReadWriteOnce . It should define the StorageClass name exam for the PersistentVolume , which will be used to bind PersistentVolumeClaim requests to this PersistenetVolume.
* Create a PefsissentVolumeClaim 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 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

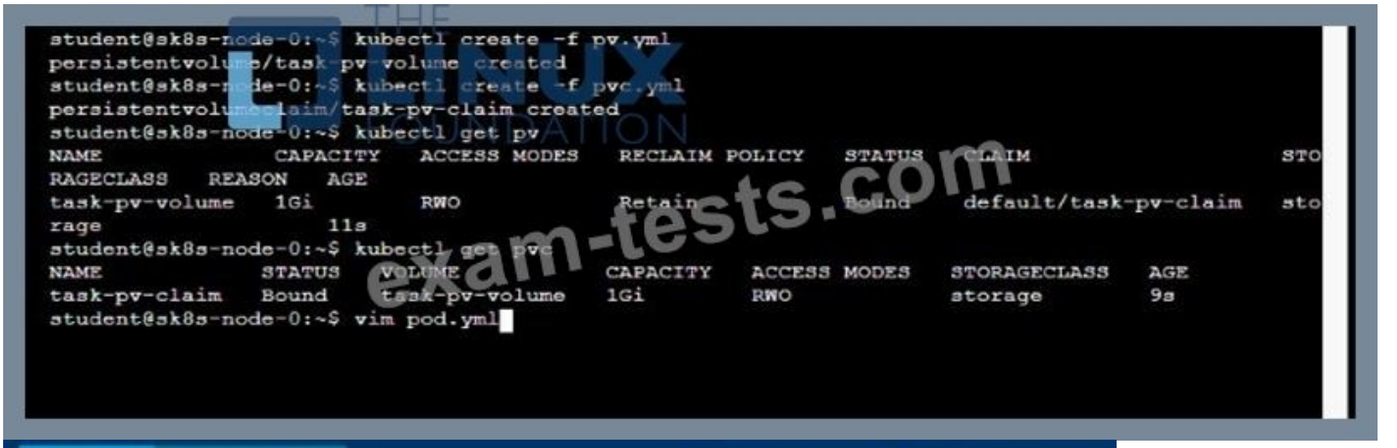
A. Solution:


Readme
Web Terminal
THE LINUX FOUNDATION
student@node-1:~$ kubectl config use-context sk8s
Switched to context "sk8s".
student@node-1:~$


Readme
Web Terminal
THE LINUX FOUNDATION
* 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
Usage of /: 10.3% of 242.29GB
Memory usage: 2%
Swap usage: 0%
Processes: 38
Users logged in:
IP address for eth0: 10.250.3.115
IP address for docker0: 172.17.0.1
IP address for cni0: 10.244.1.1
* 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:~$


Readme
Web Terminal
THE LINUX FOUNDATION
student@sk8s-node-0:~$ echo 'Acct=Finance' > /opt/KDSP00101/data/index.html
student@sk8s-node-0:~$ vim pv.yml



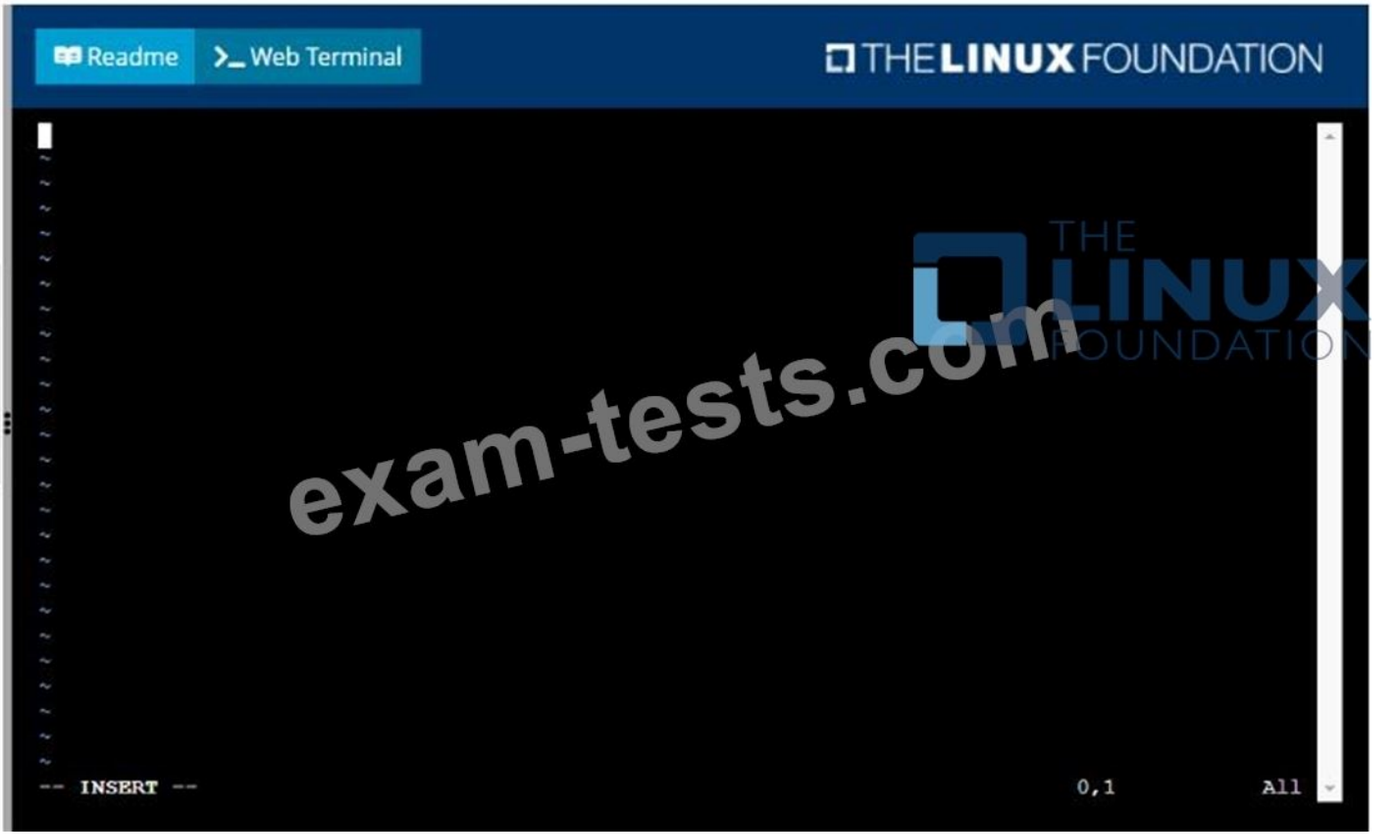

student@sk8s-node-0:~$ kubectl create -f pv.yml
persistentvolume/task-pv-volume created
student@sk8s-node-0:~$ kubectl create -f pvc.yml
persistentvolumeclaim/task-pv-claim created
student@sk8s-node-0:~$ kubectl get pv
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STO
RAGECLASS REASON AGE
task-pv-volume 1Gi RWO Retain Bound default/task-pv-claim sto
rage 11s
student@sk8s-node-0:~$ kubectl get pvc
NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE
task-pv-claim Bound task-pv-volume 1Gi RWO storage 9s
student@sk8s-node-0:~$ vim pod.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-cla
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:~$

B. Solution:


Readme
Web Terminal
THE LINUX FOUNDATION
student@node-1:~$ kubectl config use-context sk8s
Switched to context "sk8s".
student@node-1:~$



Readme | Web Terminal — THE LINUX FOUNDATION

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: task-pv-volume
spec:
  capacity:
    storage: 1Gi
  accessModes:
  - ReadWriteOnce
  storageClassName: storage
  hostPath:
    path: /opt/KDSP00101/data
    type: Directory
~
```

```
student@sk8s-node-0:~$ kubectl create -f pv.yml
persistentvolume/task-pv-volume created
student@sk8s-node-0:~$ kubectl create -f pvc.yml
persistentvolumeclaim/task-pv-claim created
student@sk8s-node-0:~$ kubectl get pv
NAME             CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                   STO
RAGECLASS   REASON   AGE
task-pv-volume   1Gi        RWO            Retain           Bound    default/task-pv-claim   sto
rage                 11s
student@sk8s-node-0:~$ kubectl get pvc
NAME            STATUS   VOLUME           CAPACITY   ACCESS MODES   STORAGECLASS   AGE
task-pv-claim   Bound    task-pv-volume   1Gi        RWO            storage        9s
student@sk8s-node-0:~$ vim pod.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
```

Answer: (SHOW ANSWER)

NEW QUESTION: 31

Context

Context

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

Task

Update the app-a deployment in the production namespace to run as the restrictedservice service account. The service account has already been created.

Answer:

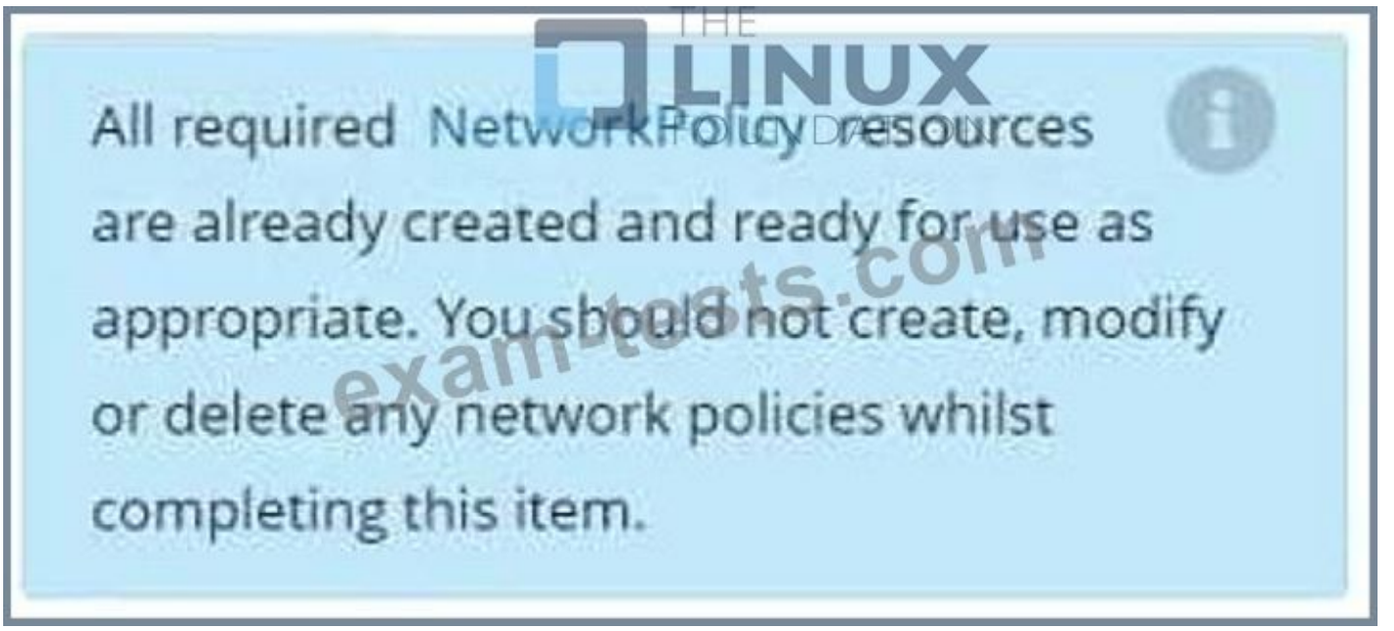
Solution:

NEW QUESTION: 32

Context

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.



Answer:

apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: internal-policy
namespace: default
spec:
podSelector:
matchLabels:
name: internal
policyTypes:
- Egress
- Ingress
ingress:
- {}
egress:
- to:
- podSelector:
matchLabels:
name: mysql
ports:
- protocol: TCP
port: 3306
- to:
- podSelector:
matchLabels:
name: payroll
ports:

- protocol: TCP
port: 8080
- ports:
- port: 53
protocol: UDP
- port: 53
protocol: TCP

NEW QUESTION: 33

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

Answer:
See the solution below.
Explanation
Solution:


Readme
Web Terminal
THE LINUX FOUNDATION
apiVersion: v1
kind: Pod
metadata:
creationTimestamp: null
labels:
run: nginx-resources
name: nginx-resources
namespace: pod-resources
spec:
containers:
- image: nginx
name: nginx-resources
resources: {}
dnsPolicy: ClusterFirst
restartPolicy: Always
status: {}
"nginx_resources.yml" 16L, 289C
1,1
All
exam-tests.com


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
exam-tests.com


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
exam-tests.com