⇨Vendor: Linux Foundation

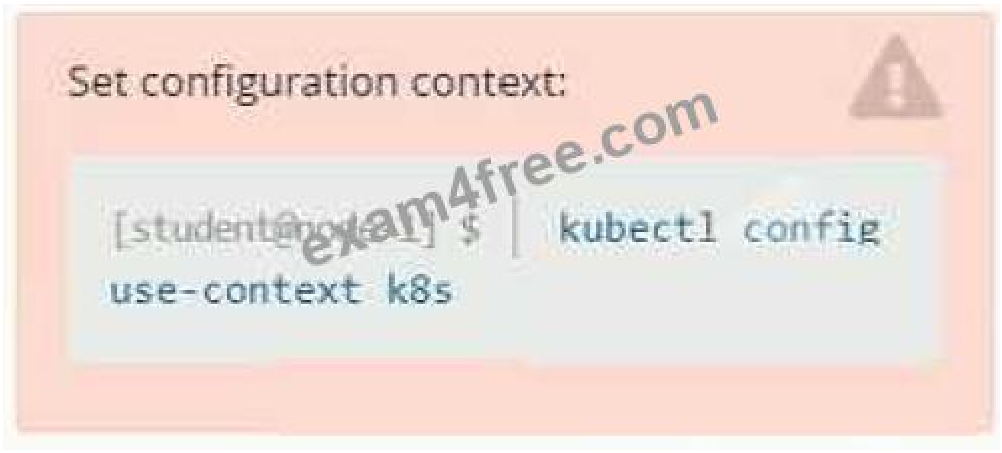
⇨Exam Code: CKAD

⇨Exam Name: Linux Foundation Certified Kubernetes Application Developer Exam

New Updated Questions from Exam4free (Updated in Jan, 2023)

Linux Foundation

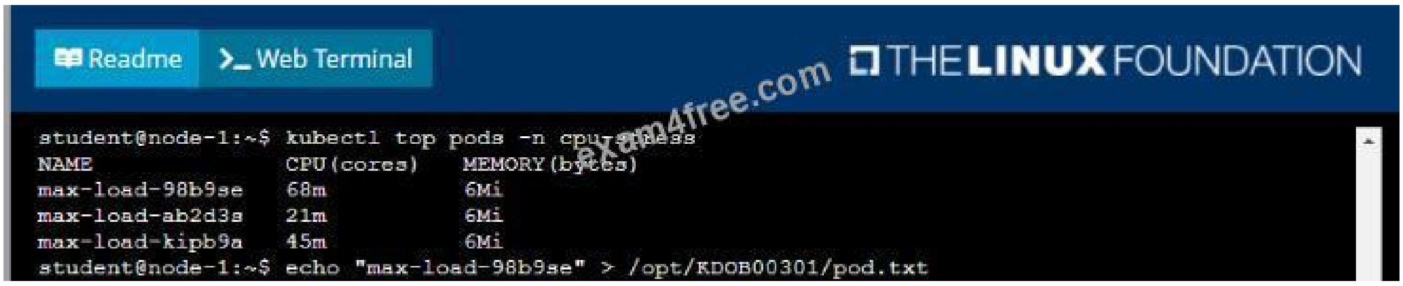
NEW QUESTION 13



Set configuration context:

```
[student@node-1] $ | kubectl config
use-context k8s
```

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:
Explanation:
See the solution below.
Explanation
Solution:

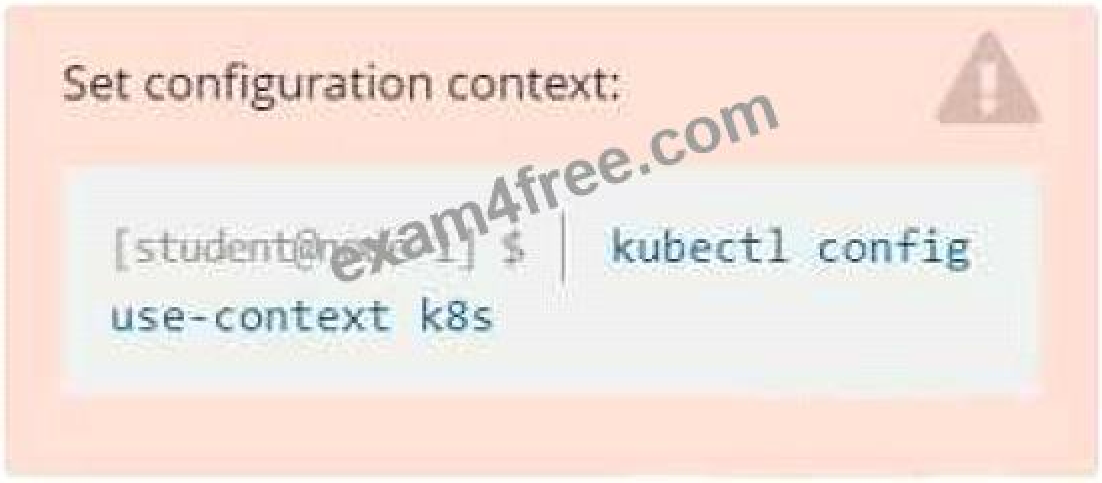


Readme >_ Web Terminal THE LINUX FOUNDATION

```
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/KDOB00301/pod.txt
```

NEW QUESTION 14

Context



Set configuration context:

```
[student@node-1] $ | kubectl config
use-context k8s
```

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:

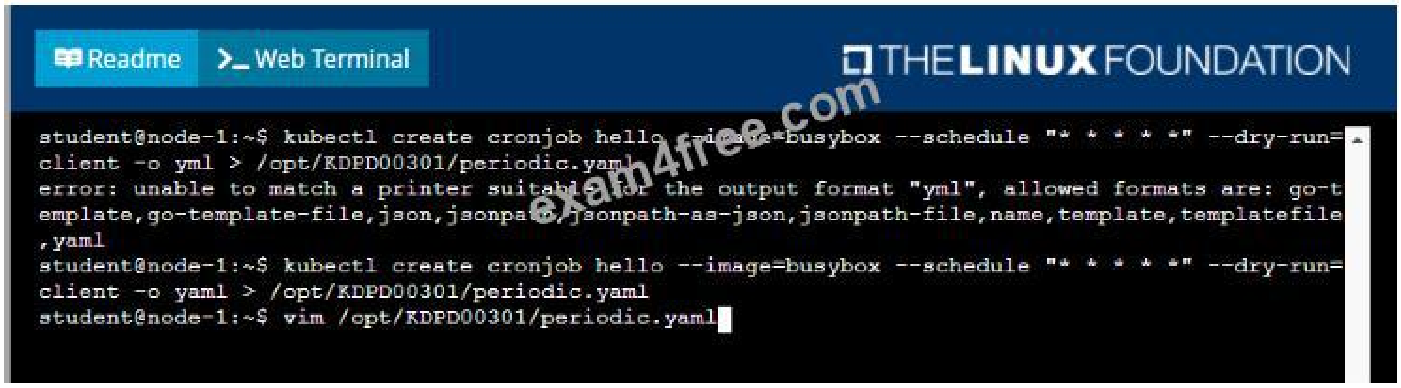
* Create a YAML formatted Kubernetes manifest /opt/KDPD00301/periodic.yaml that runs the following shell command: date in a single busybox container. The command should run every minute and must complete within 22 seconds or be terminated oy Kubernetes. The Cronjob namp and container name should both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

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

Answer:

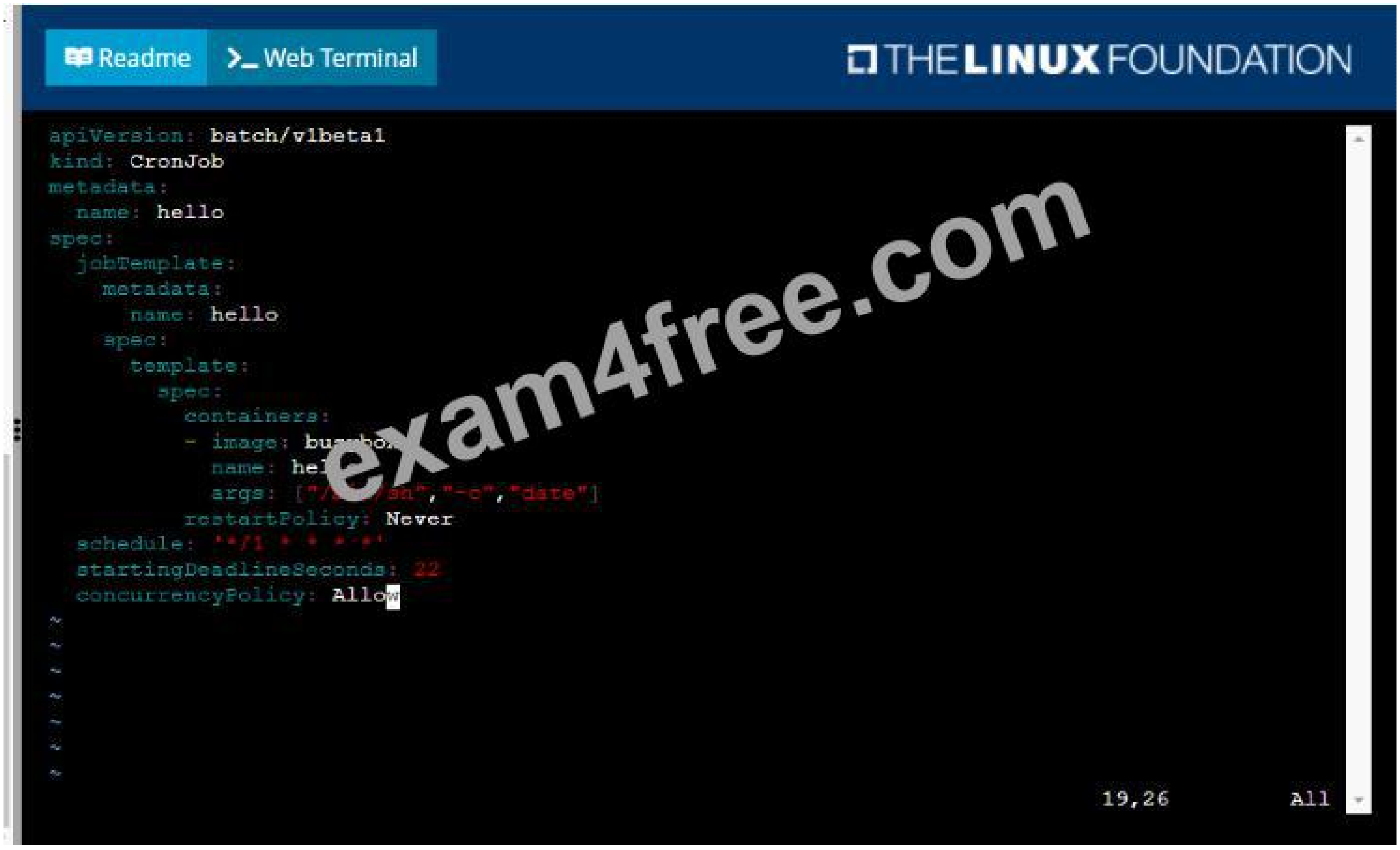
Explanation:

Solution:

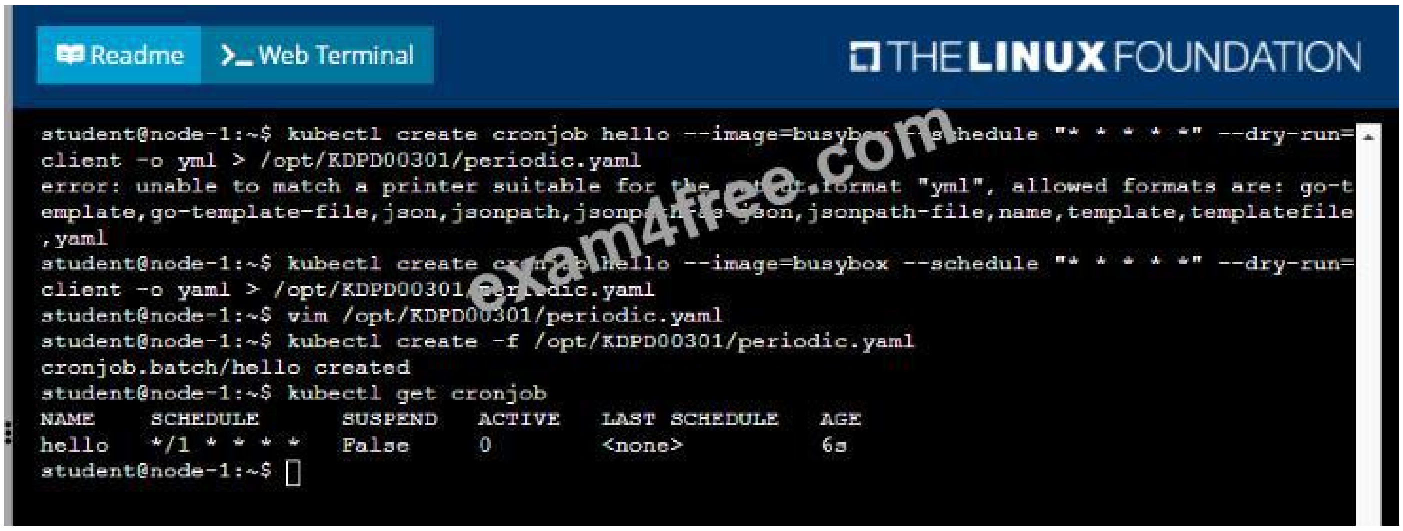


Readme >_ Web Terminal THE LINUX FOUNDATION

```
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yml > /opt/KDPD00301/periodic.yaml
error: unable to match a printer suitable for the output format "yml", allowed formats are: go-t
emplate,go-template-file,json,jsonpath,jsonpath-as-json,jsonpath-file,name,template,templatefile
,yaml
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yaml > /opt/KDPD00301/periodic.yaml
student@node-1:~$ vim /opt/KDPD00301/periodic.yaml
```

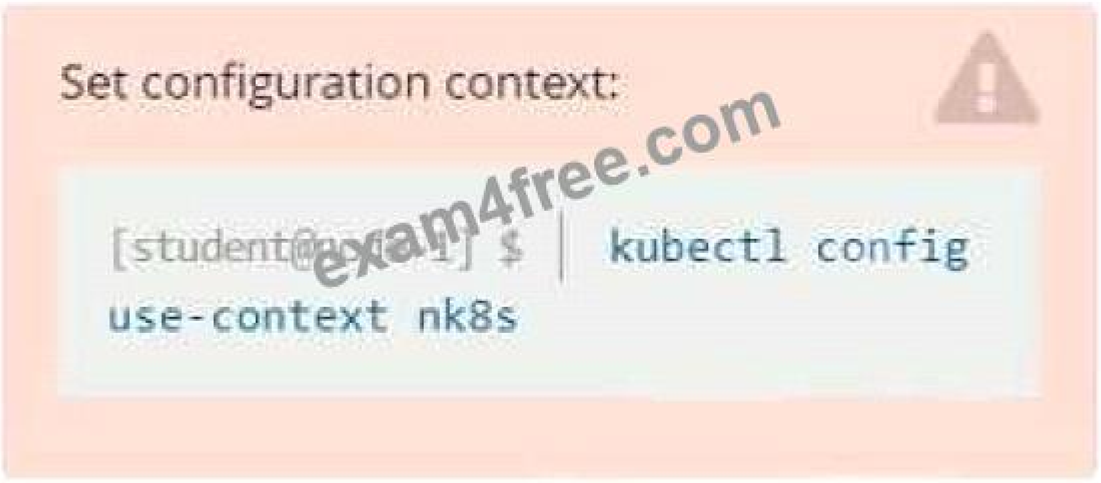


```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: hello
spec:
  jobTemplate:
    metadata:
      name: hello
    spec:
      template:
        spec:
          containers:
          - image: busybox
            name: hello
            args: ["/bin/sh","-c","date"]
          restartPolicy: Never
  schedule: '*/1 * * * *'
  startingDeadlineSeconds: 22
  concurrencyPolicy: Allow
~
~
~
~
~
~
~
                                                       19,26         All
```



```
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yml > /opt/KDPD00301/periodic.yaml
error: unable to match a printer suitable for the output format "yml", allowed formats are: go-t
emplate,go-template-file,json,jsonpath,jsonpath-as-json,jsonpath-file,name,template,templatefile
,yaml
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yaml > /opt/KDPD00301/periodic.yaml
student@node-1:~$ vim /opt/KDPD00301/periodic.yaml
student@node-1:~$ kubectl create -f /opt/KDPD00301/periodic.yaml
cronjob.batch/hello created
student@node-1:~$ kubectl get cronjob
NAME    SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
hello   */1 * * * *   False     0        <none>          6s
student@node-1:~$ 
```

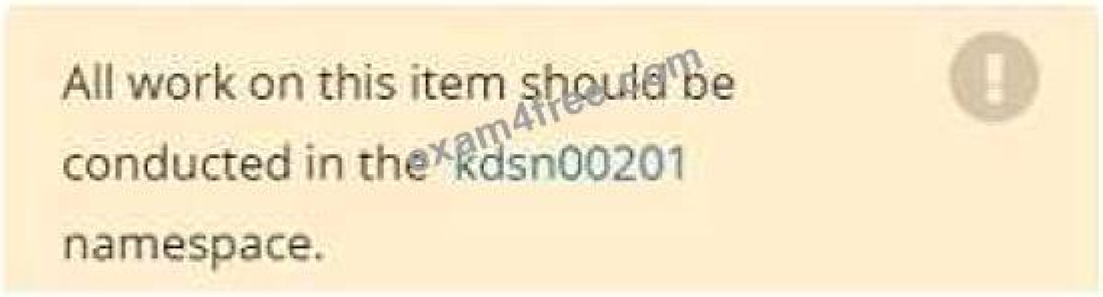
NEW QUESTION 15
Exhibit:



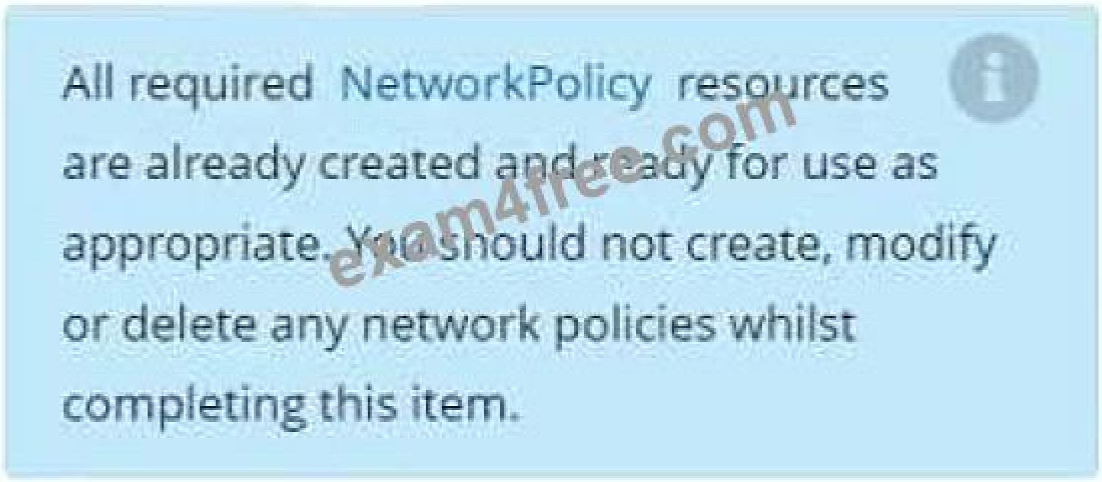
Set configuration context:

```
[student@node-1] $ | kubectl config
use-context nk8s
```

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.



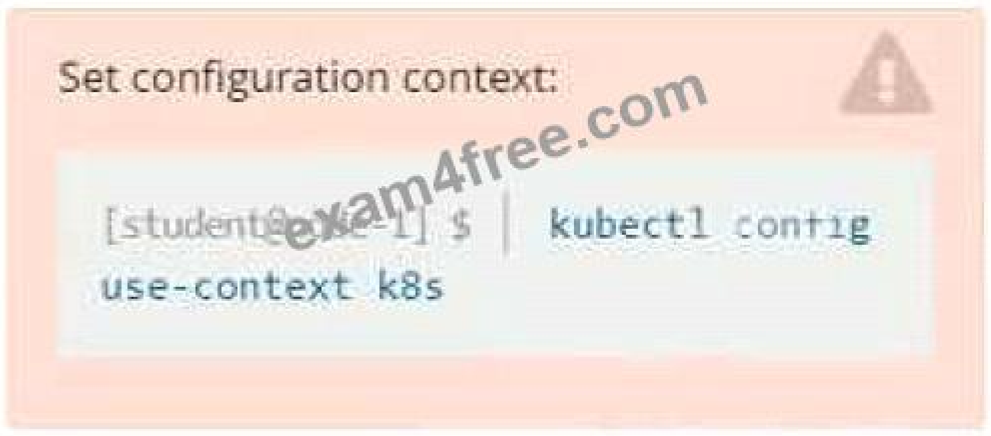
All work on this item should be conducted in the kdsn00201 namespace.



All required NetworkPolicy resources are already created and ready for use as appropriate. You should not create, modify or delete any network policies whilst completing this item.

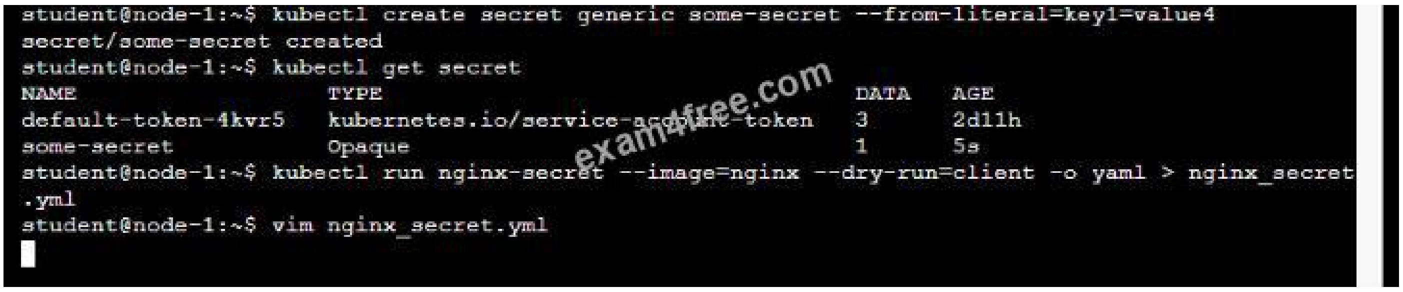
A. Pending
Answer: A

NEW QUESTION 16
Exhibit:

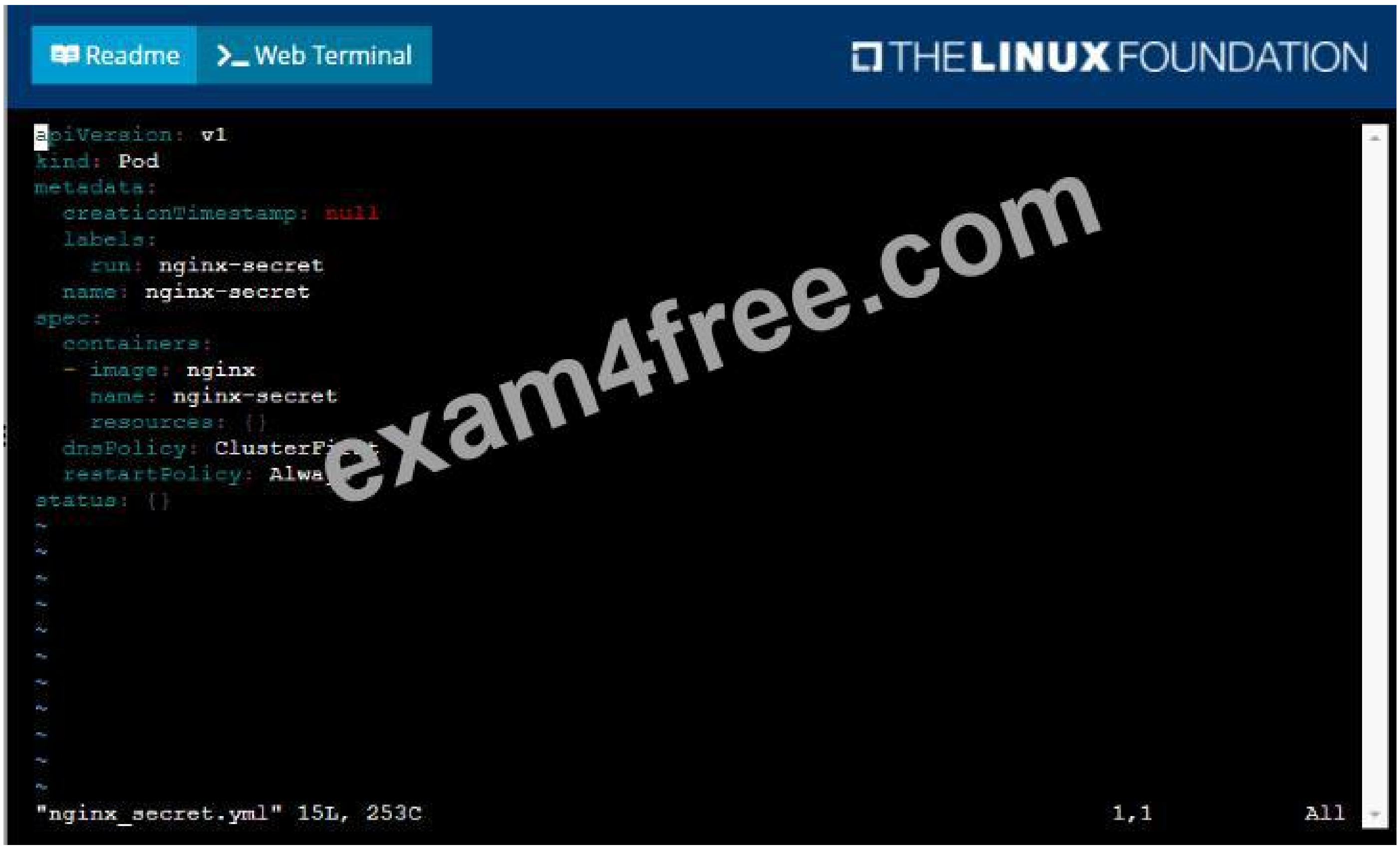

Set configuration context:

```
[student@node-1] $ | kubectl config
use-context k8s
```

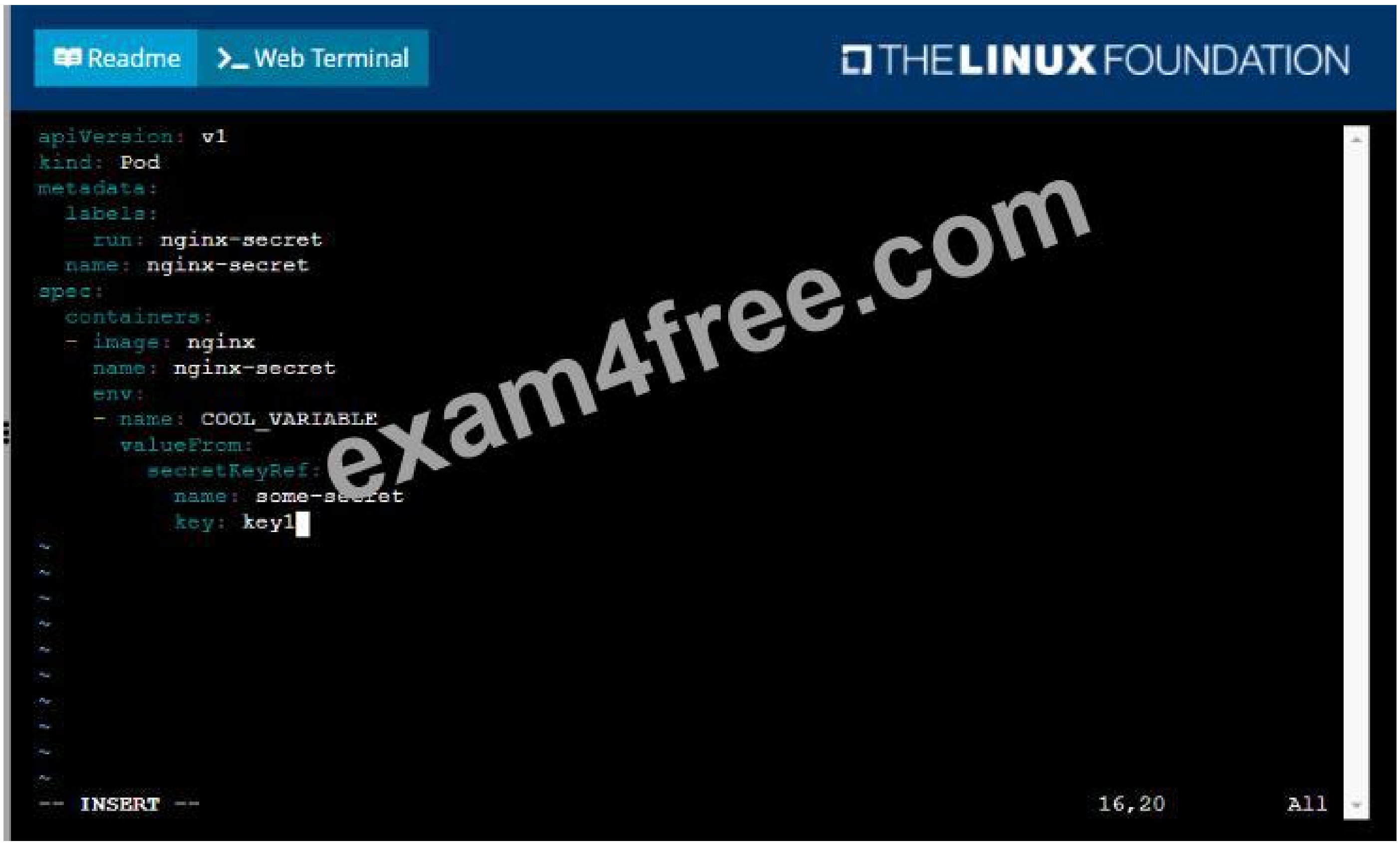
Context
You are tasked to create a secret and consume the secret in a pod using environment variables as follow:
Task
* Create a secret named another-secret with a key/value pair; key1/value4
* Start an nginx pod named nginx-secret using container image nginx, and add an environment variable exposing the value of the secret key key 1, using COOL_VARIABLE as the name for the environment variable inside the pod
A. Solution:



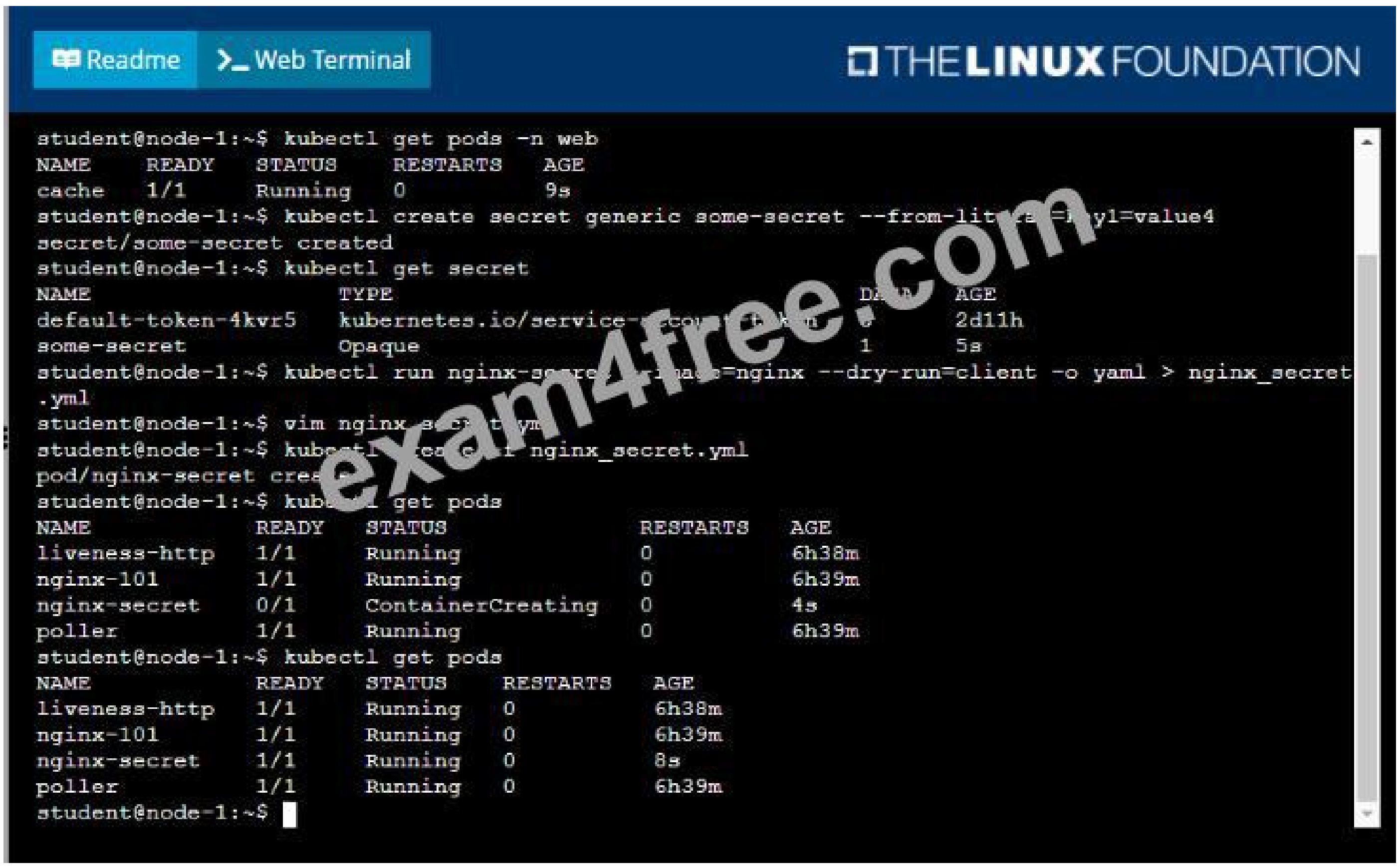
```
student@node-1:~$ kubectl create secret generic some-secret --from-literal=key1=value4
secret/some-secret created
student@node-1:~$ kubectl get secret
NAME                  TYPE                                  DATA   AGE
default-token-4kvr5   kubernetes.io/service-account-token   3      2d11h
some-secret           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
```



```
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
```

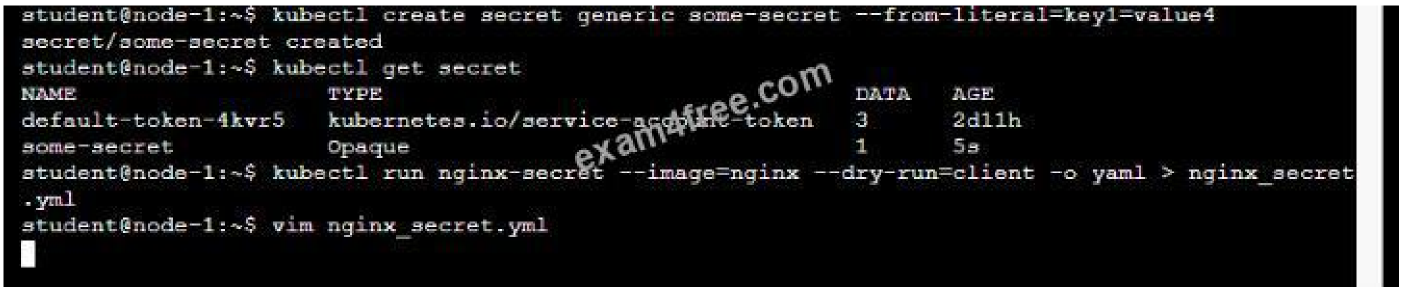


```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx-secret
  name: nginx-secret
spec:
  containers:
  - image: nginx
    name: nginx-secret
    env:
    - name: COOL_VARIABLE
      valueFrom:
        secretKeyRef:
          name: some-secret
          key: key1
~
~
~
~
~
~
~
~
~
~
-- INSERT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

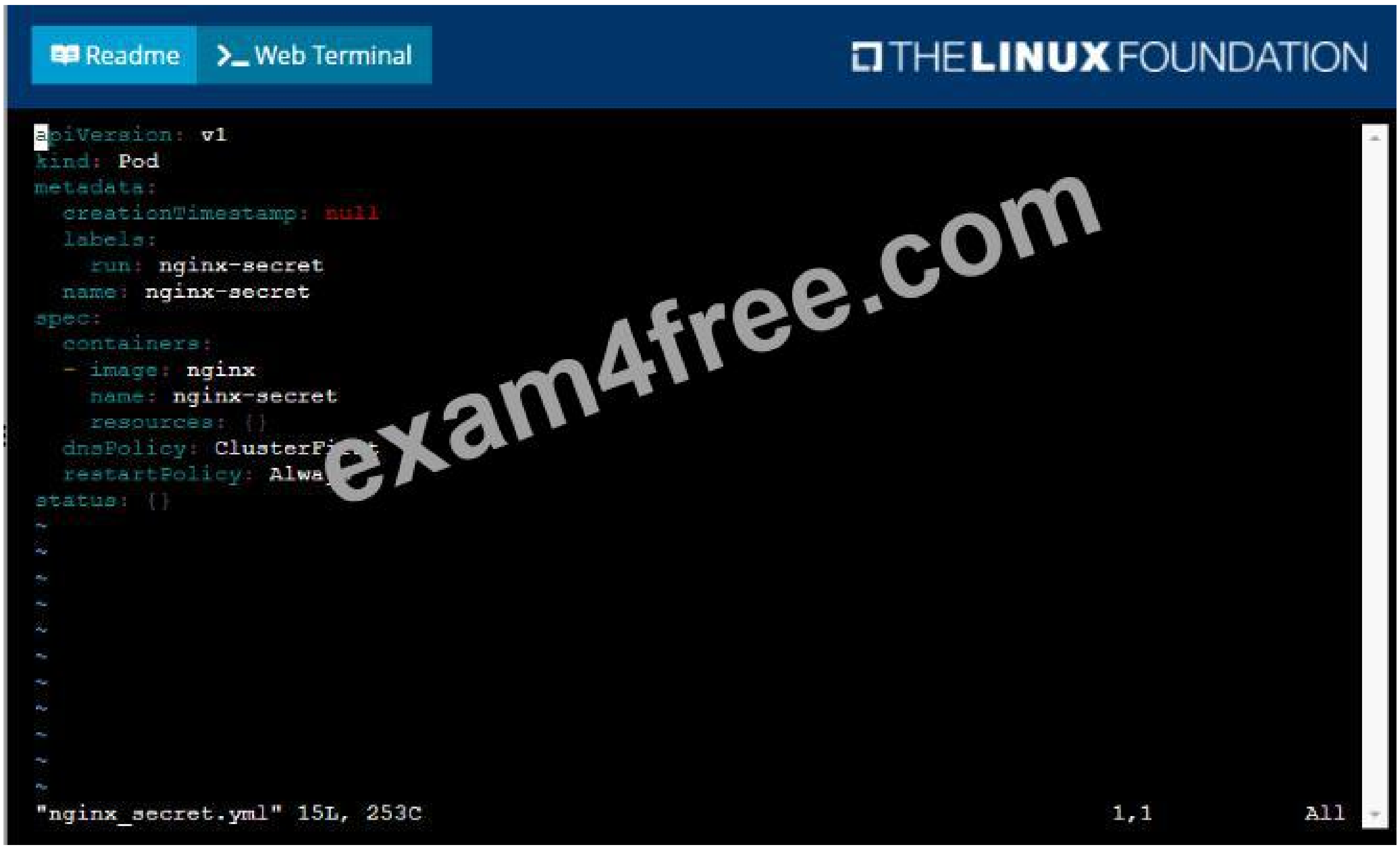


```
student@node-1:~$ kubectl get pods -n web
NAME     READY   STATUS    RESTARTS   AGE
cache    1/1     Running   0          9s
student@node-1:~$ kubectl create secret generic some-secret --from-literal=key1=value4
secret/some-secret created
student@node-1:~$ kubectl get secret
NAME                  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:~$
```

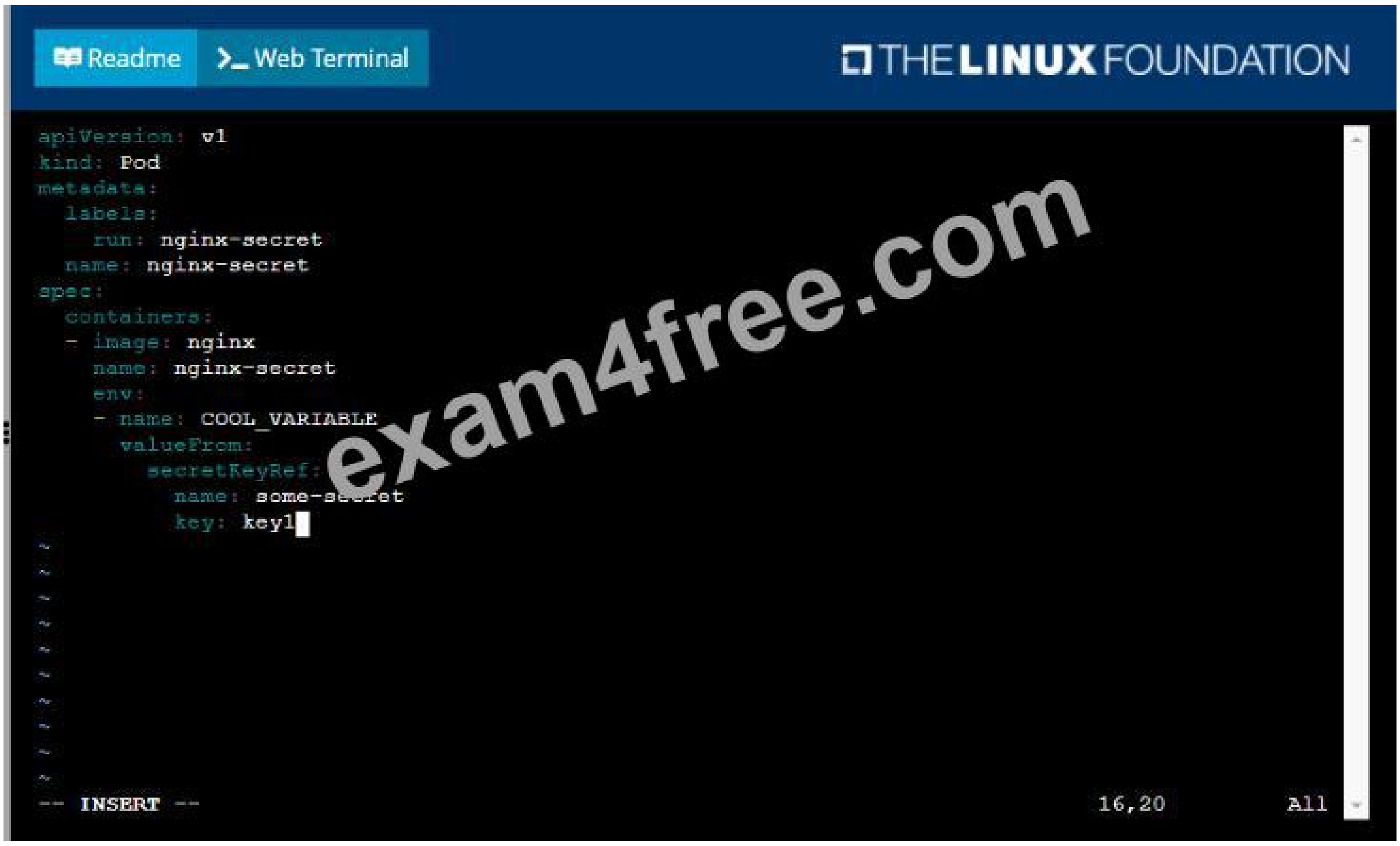
B. Solution:



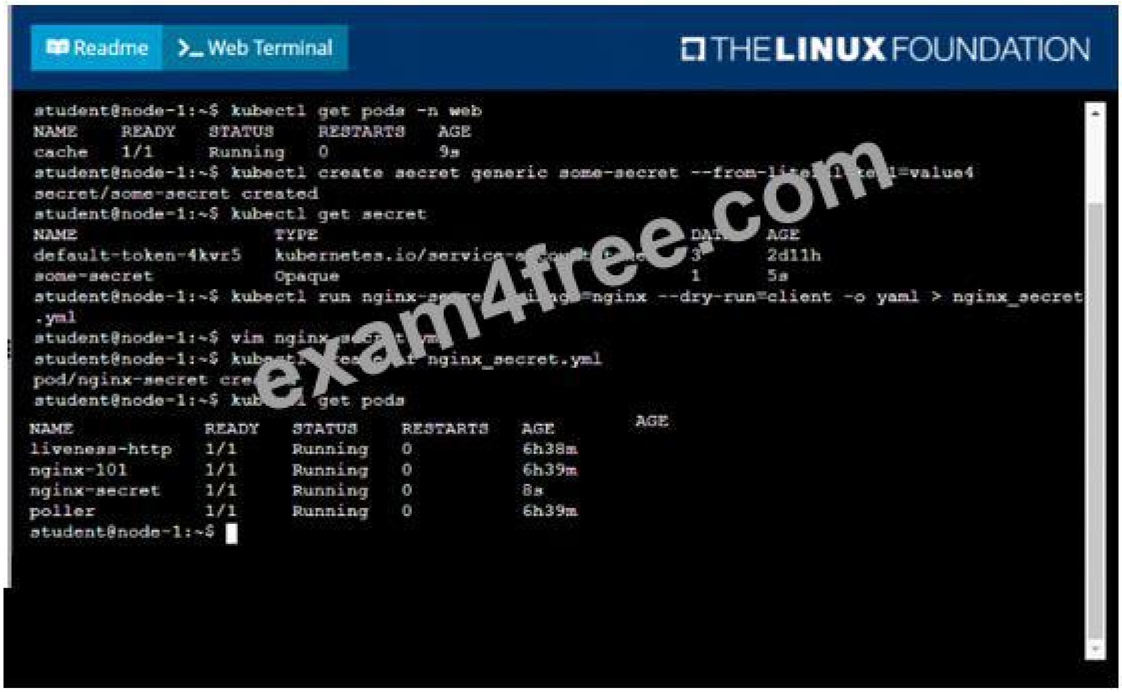
```
student@node-1:~$ kubectl create secret generic some-secret --from-literal=key1=value4
secret/some-secret created
student@node-1:~$ kubectl get secret
NAME                  TYPE                                  DATA   AGE
default-token-4kvr5   kubernetes.io/service-account-token   3      2d11h
some-secret           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
```



```
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
```



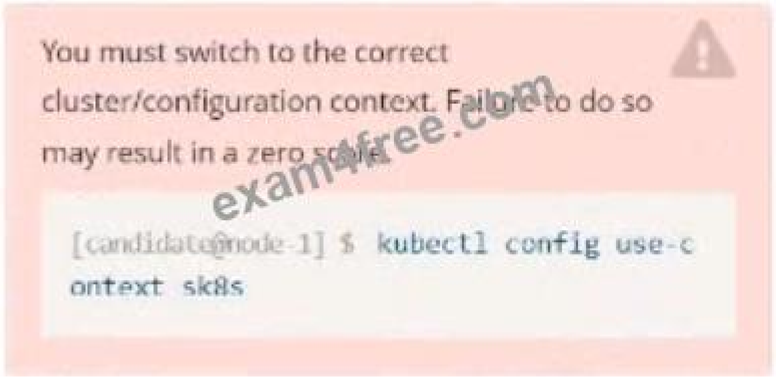
```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx-secret
  name: nginx-secret
spec:
  containers:
  - image: nginx
    name: nginx-secret
    env:
    - name: COOL_VARIABLE
      valueFrom:
        secretKeyRef:
          name: some-secret
          key: key1
~
~
~
~
~
~
~
~
~
~
-- INSERT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



```
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
                                                         AGE
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:~$
```

Answer: A

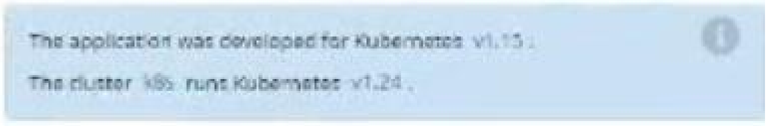
NEW QUESTION 17



You must switch to the correct cluster/configuration context. Failure to do so may result in a zero score.

```
[candidate@node-1] $ kubectl config use-context sk8s
```

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.



The application was developed for Kubernetes v1.15 .
The cluster k8s runs Kubernetes v1.24 .

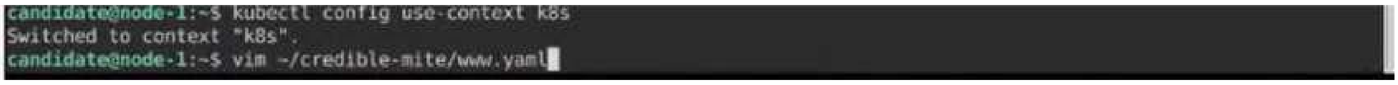
2) Deploy the application specified in the updated manifest file -/credible-mite/www.yaml in namespace cobra See the solution below.

Answer:

Explanation:

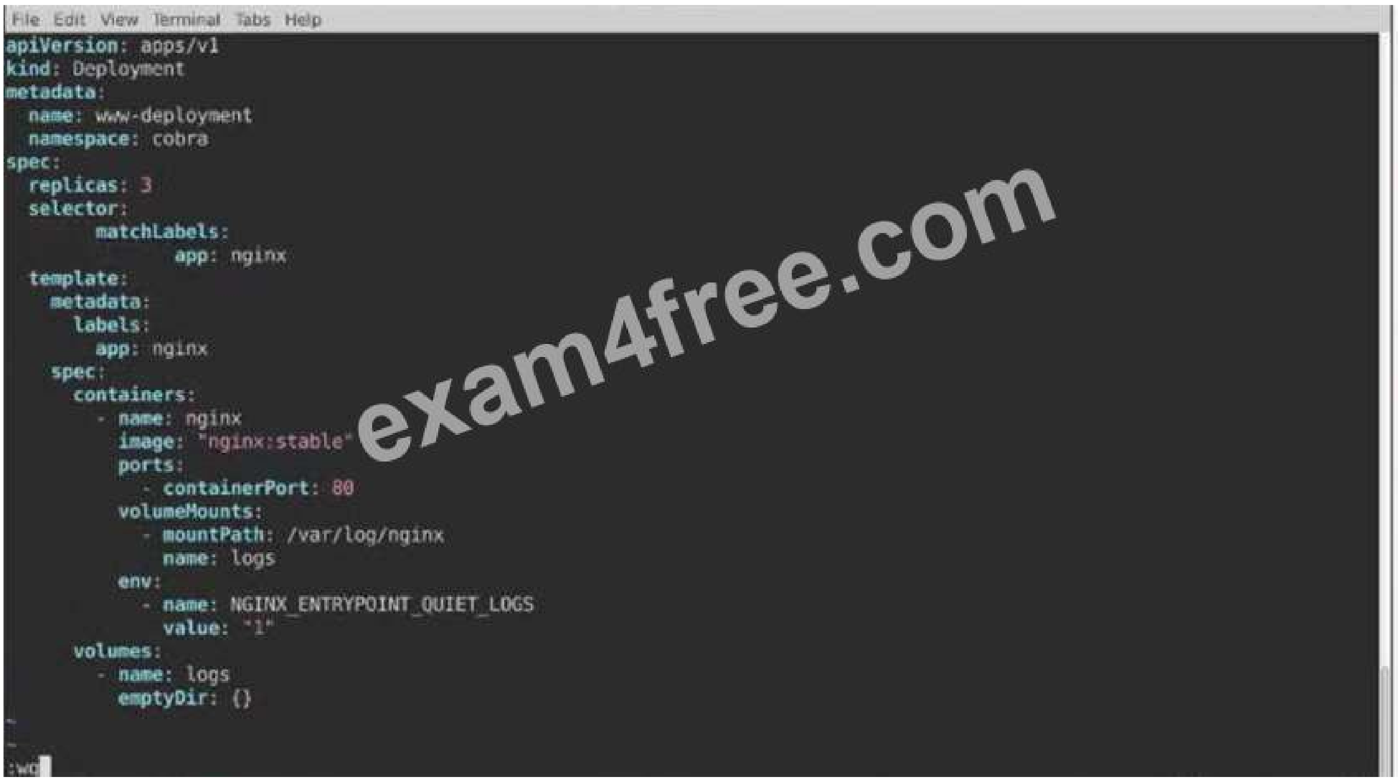
Explanation

Solution:



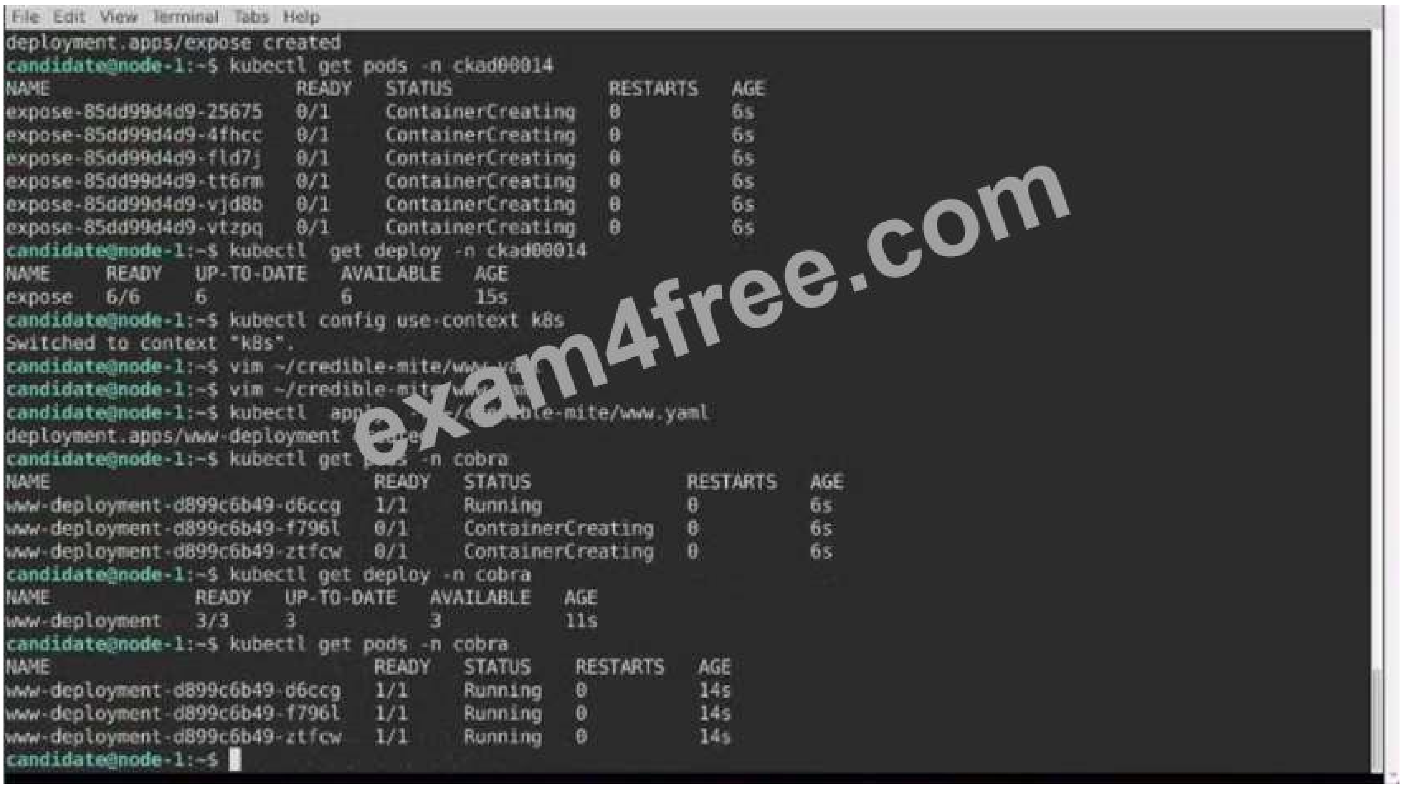
```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/credible-mite/www.yaml
```

Text Description automatically generated



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: www-deployment
  namespace: cobra
spec:
  replicas: 3
  selector:
        matchLabels:
            app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: "nginx:stable"
          ports:
            - containerPort: 80
          volumeMounts:
            - mountPath: /var/log/nginx
              name: logs
          env:
            - name: NGINX_ENTRYPOINT_QUIET_LOGS
              value: "1"
      volumes:
        - name: logs
          emptyDir: {}
~
~
:wq
```

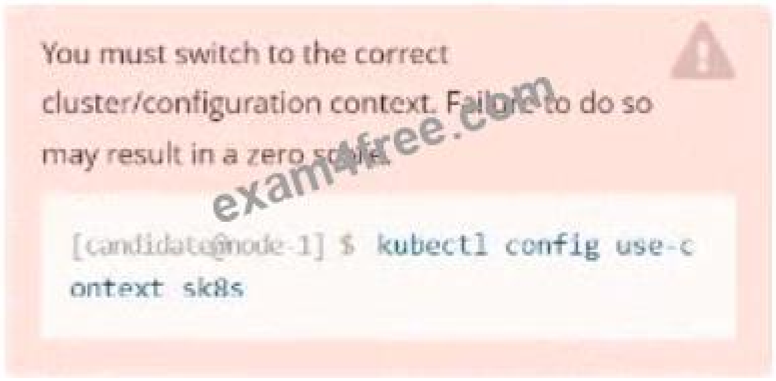
Text Description automatically generated



```
deployment.apps/expose created
candidate@node-1:~$ kubectl get pods -n ckad00014
NAME                        READY   STATUS              RESTARTS   AGE
expose-85dd99d4d9-25675     0/1     ContainerCreating   0          6s
expose-85dd99d4d9-4fhcc     0/1     ContainerCreating   0          6s
expose-85dd99d4d9-fld7j     0/1     ContainerCreating   0          6s
expose-85dd99d4d9-tt6rm     0/1     ContainerCreating   0          6s
expose-85dd99d4d9-vjd8b     0/1     ContainerCreating   0          6s
expose-85dd99d4d9-vtzpq     0/1     ContainerCreating   0          6s
candidate@node-1:~$ kubectl  get deploy -n ckad00014
NAME     READY   UP-TO-DATE   AVAILABLE   AGE
expose   6/6     6            6           15s
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/credible-mite/www.yaml
candidate@node-1:~$ vim ~/credible-mite/www.yaml
candidate@node-1:~$ kubectl  apply -f ~/credible-mite/www.yaml
deployment.apps/www-deployment created
candidate@node-1:~$ kubectl get pods -n cobra
NAME                              READY   STATUS              RESTARTS   AGE
www-deployment-d899c6b49-d6ccg    1/1     Running             0          6s
www-deployment-d899c6b49-f796l    0/1     ContainerCreating   0          6s
www-deployment-d899c6b49-ztfcw    0/1     ContainerCreating   0          6s
candidate@node-1:~$ kubectl get deploy -n cobra
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
www-deployment   3/3     3            3           11s
candidate@node-1:~$ kubectl get pods -n cobra
NAME                              READY   STATUS    RESTARTS   AGE
www-deployment-d899c6b49-d6ccg    1/1     Running   0          14s
www-deployment-d899c6b49-f796l    1/1     Running   0          14s
www-deployment-d899c6b49-ztfcw    1/1     Running   0          14s
candidate@node-1:~$
```

NEW QUESTION 18

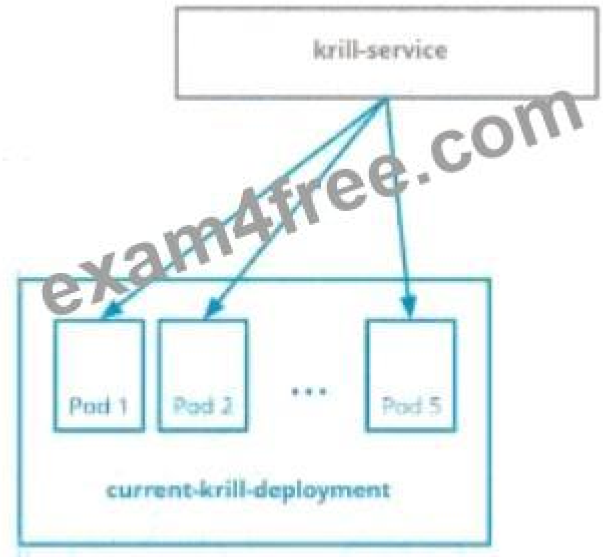
Context



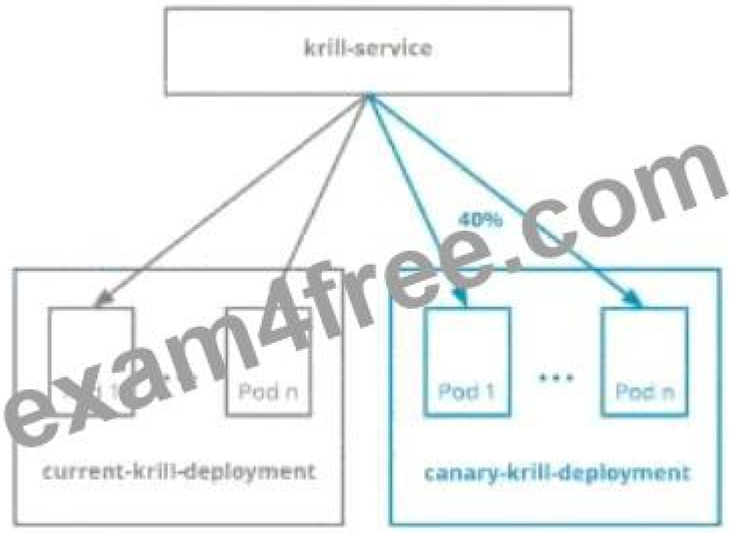
You must switch to the correct cluster/configuration context. Failure to do so may result in a zero score.

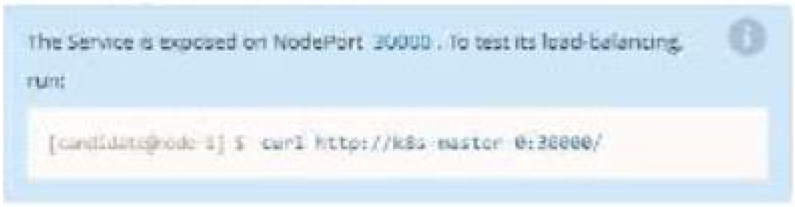
```
[candidate@node-1] $ kubectl config use-c
ontext sk8s
```

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)



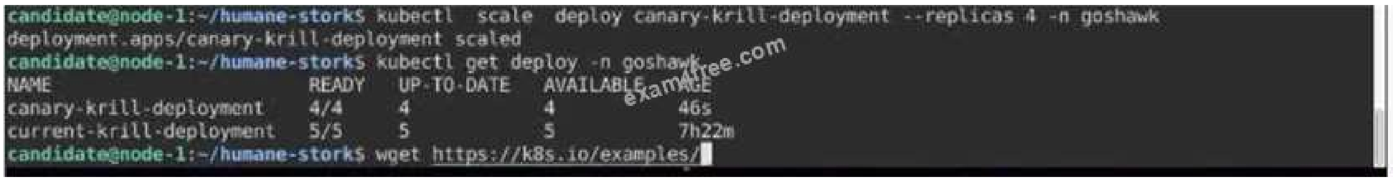

The Service is exposed on NodePort 30000 . To test its load-balancing, run:

```
[candidate@node-1] $ curl http://k8s-master-0:30000/
```

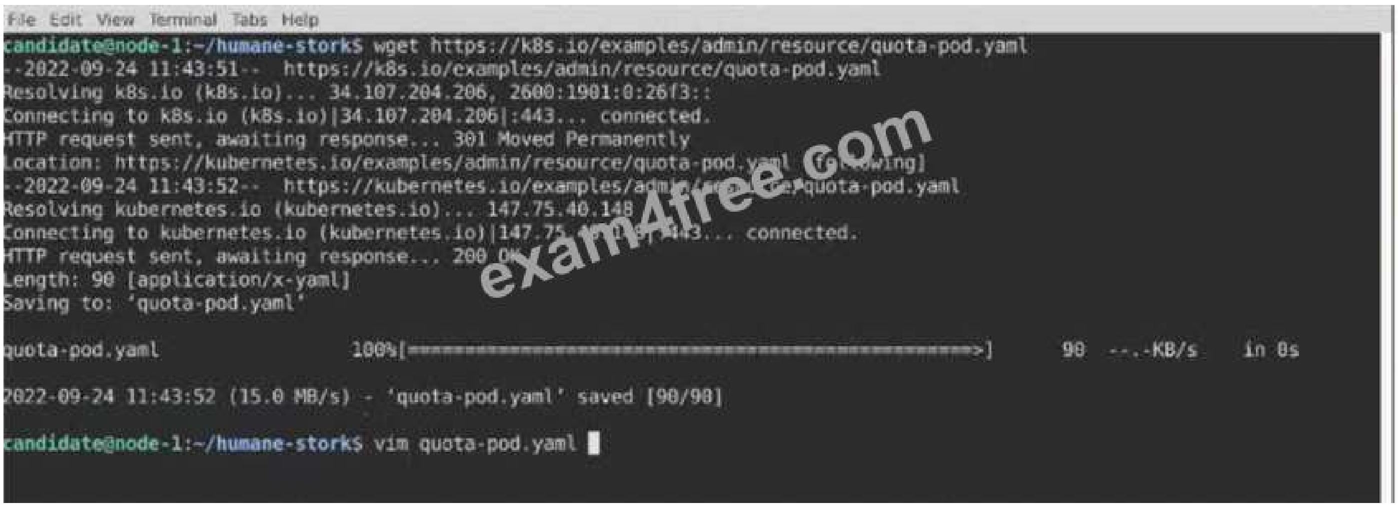
Answer:

Explanation:

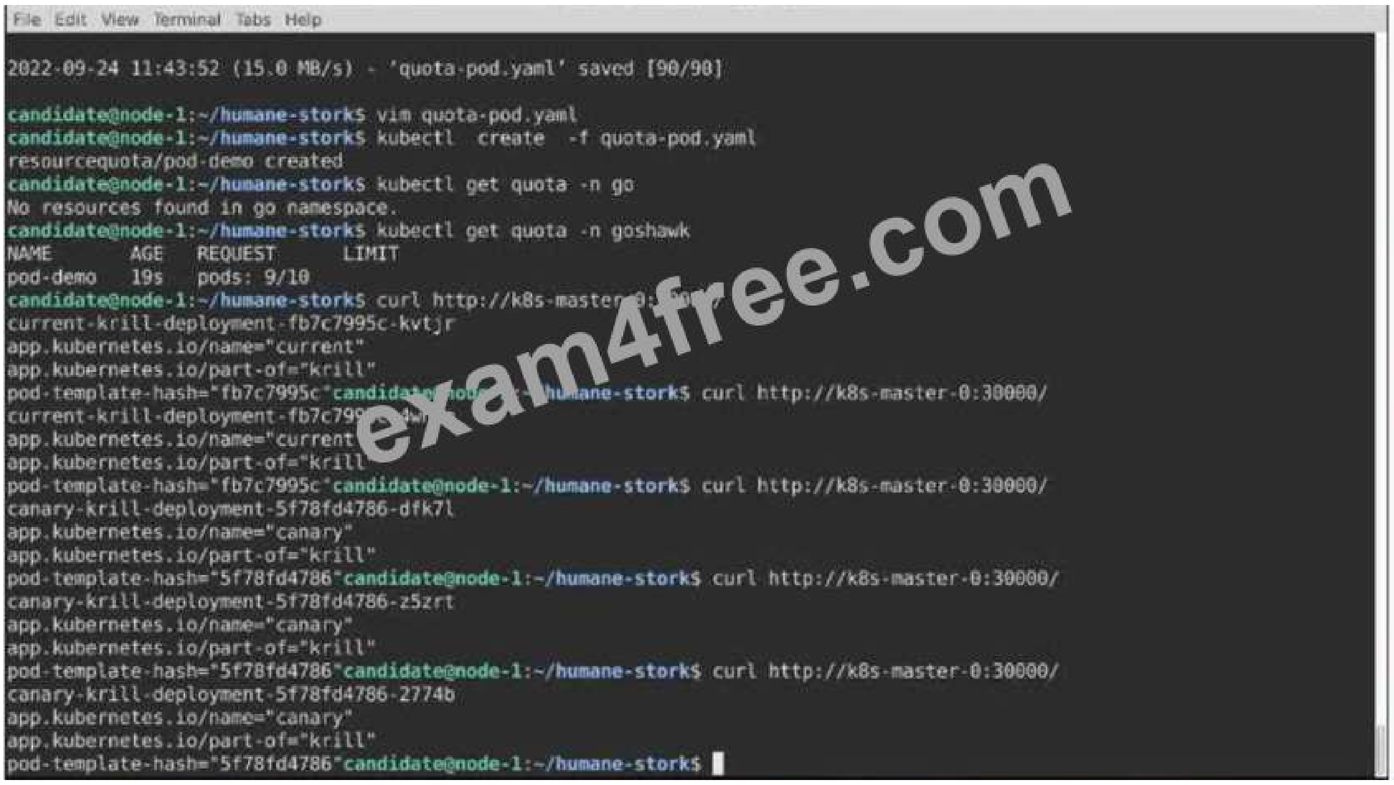
Solution:



```
candidate@node-1:~/humane-storks$ kubectl  scale  deploy canary-krill-deployment --replicas 4 -n goshawk
deployment.apps/canary-krill-deployment scaled
candidate@node-1:~/humane-storks$ 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-storks$ wget https://k8s.io/examples/
```

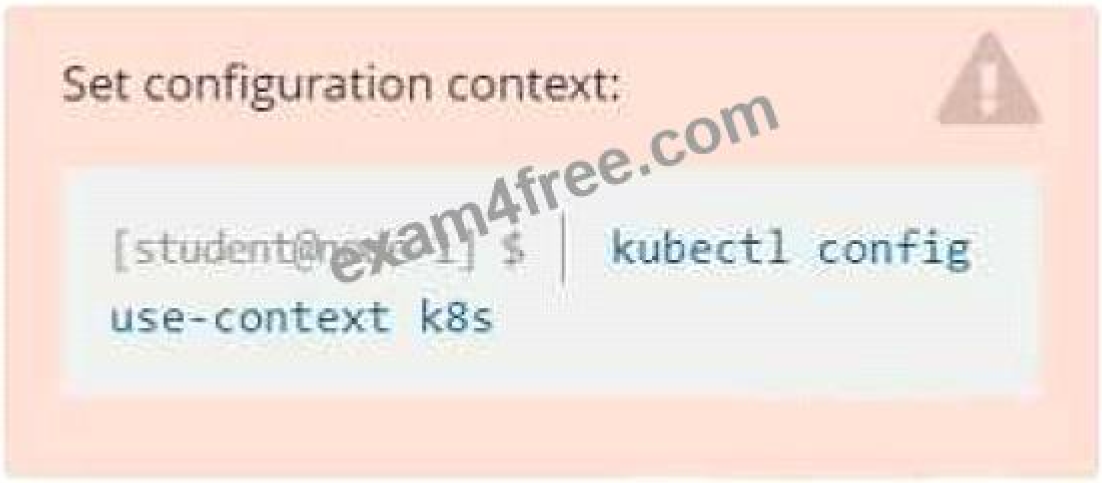


```
candidate@node-1:~/humane-storks$ wget https://k8s.io/examples/admin/resource/quota-pod.yaml
--2022-09-24 11:43:51--  https://k8s.io/examples/admin/resource/quota-pod.yaml
Resolving k8s.io (k8s.io)... 34.107.204.206, 2600:1901:0:26f3::
Connecting to k8s.io (k8s.io)|34.107.204.206|:443... connected.
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
```



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

candidate@node-1:~/humane-stork$ vim quota-pod.yaml
candidate@node-1:~/humane-stork$ kubectl  create  -f quota-pod.yaml
resourcequota/pod-demo created
candidate@node-1:~/humane-stork$ kubectl get quota -n go
No resources found in go namespace.
candidate@node-1:~/humane-stork$ kubectl get quota -n goshawk
NAME        AGE    REQUEST       LIMIT
pod-demo    19s    pods: 9/10
candidate@node-1:~/humane-stork$ 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-stork$ curl http://k8s-master-0:30000/
current-krill-deployment-fb7c799[illegible]4w
app.kubernetes.io/name="current"
app.kubernetes.io/part-of="krill"
pod-template-hash="fb7c7995c"candidate@node-1:~/humane-stork$ 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-stork$ 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-stork$ 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-stork$ █
```

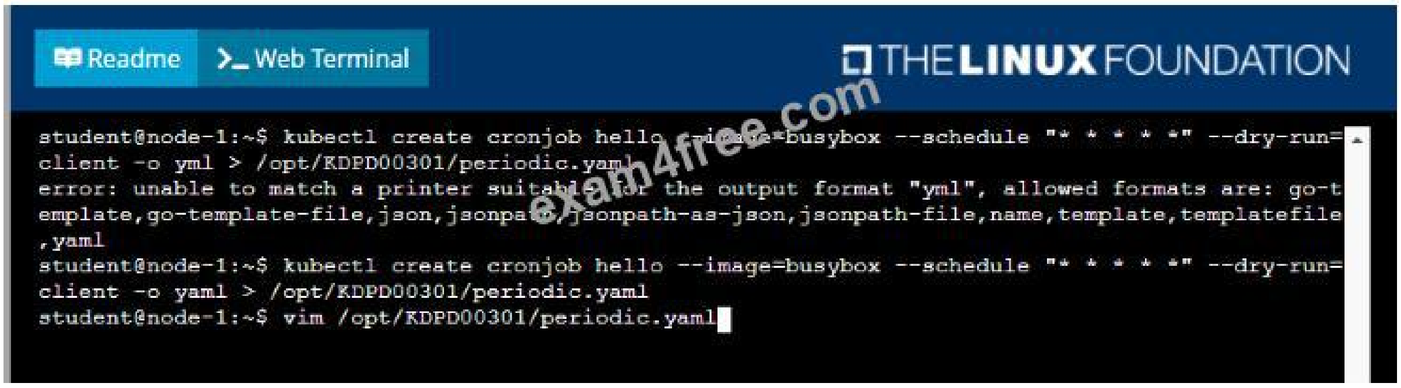
NEW QUESTION 19
Exhibit:



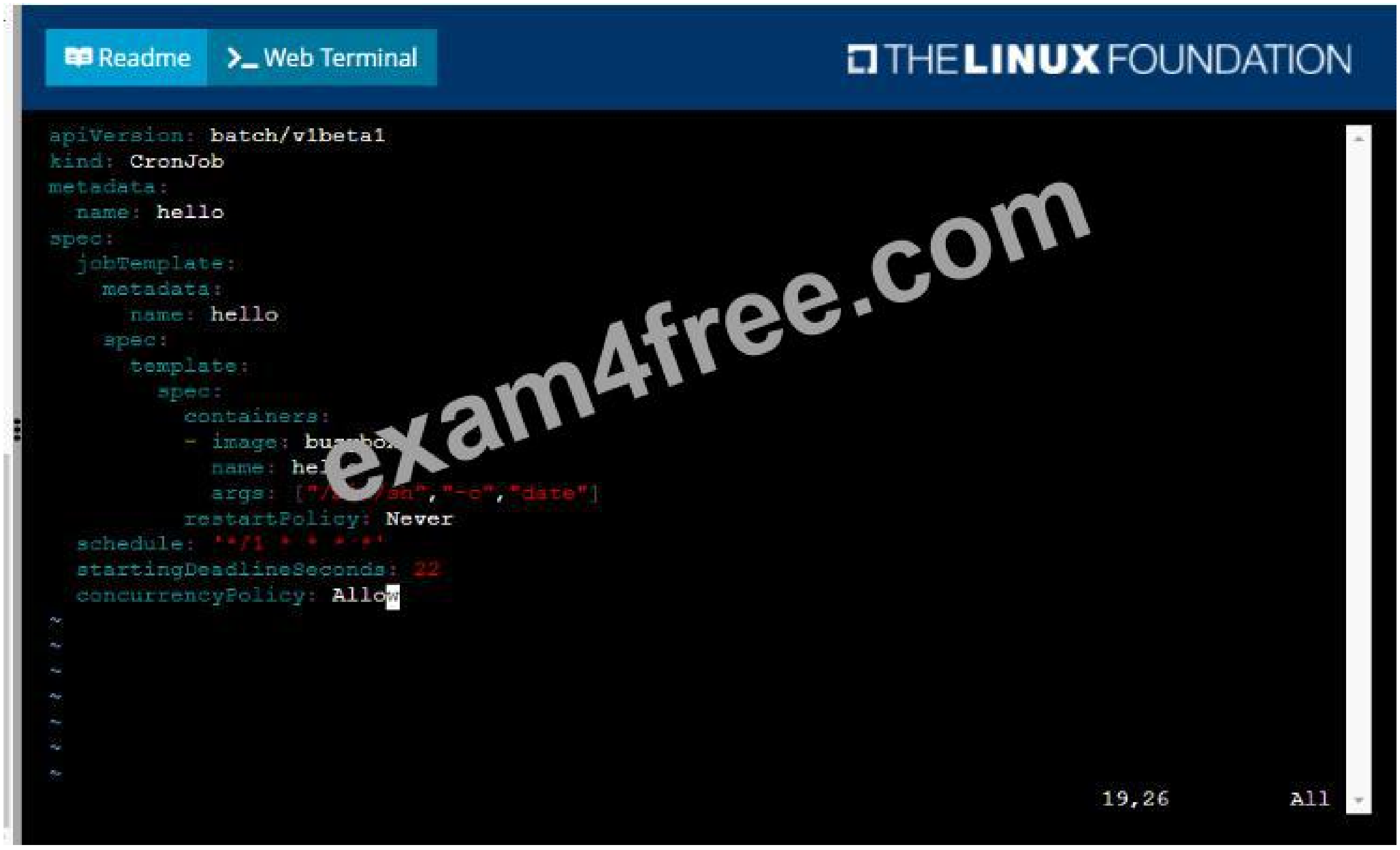
Set configuration context:

```
[student@node-1] $ | kubectl config
use-context k8s
```

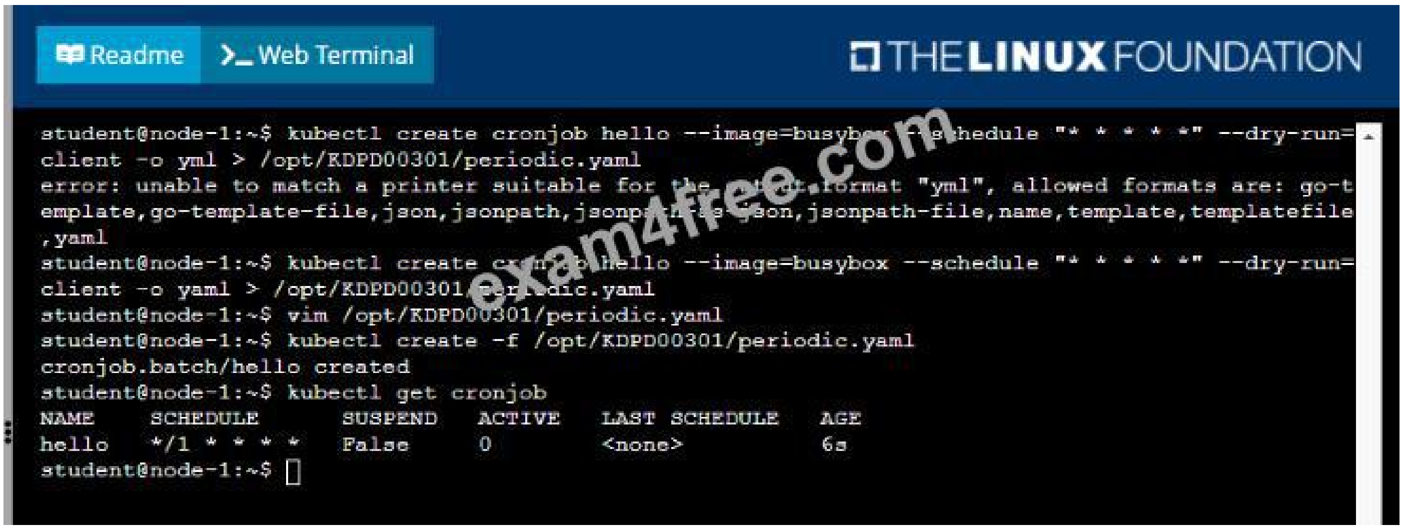
Context
Developers occasionally need to submit pods that run periodically.
Task
Follow the steps below to create a pod that will start at a predetermined time and]which runs to completion only once each time it is started:
* Create a YAML formatted Kubernetes manifest /opt/KDPD00301/periodic.yaml that 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:



```
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yml > /opt/KDPD00301/periodic.yaml
error: unable to match a printer suitable for the output format "yml", allowed formats are: go-t
emplate,go-template-file,json,jsonpath,jsonpath-as-json,jsonpath-file,name,template,templatefile
,yaml
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yaml > /opt/KDPD00301/periodic.yaml
student@node-1:~$ vim /opt/KDPD00301/periodic.yaml
```

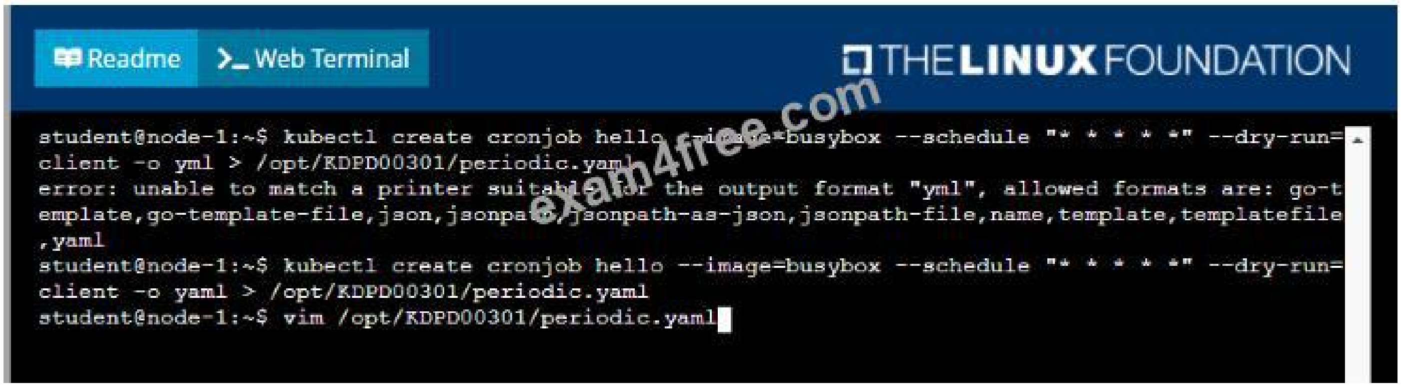


```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: hello
spec:
  jobTemplate:
    metadata:
      name: hello
    spec:
      template:
        spec:
          containers:
          - image: busybox
            name: hello
            args: ["/bin/sh","-c","date"]
          restartPolicy: Never
  schedule: '*/1 * * * *'
  startingDeadlineSeconds: 22
  concurrencyPolicy: Allow
~
~
~
~
~
~
~
                                                              19,26         All
```

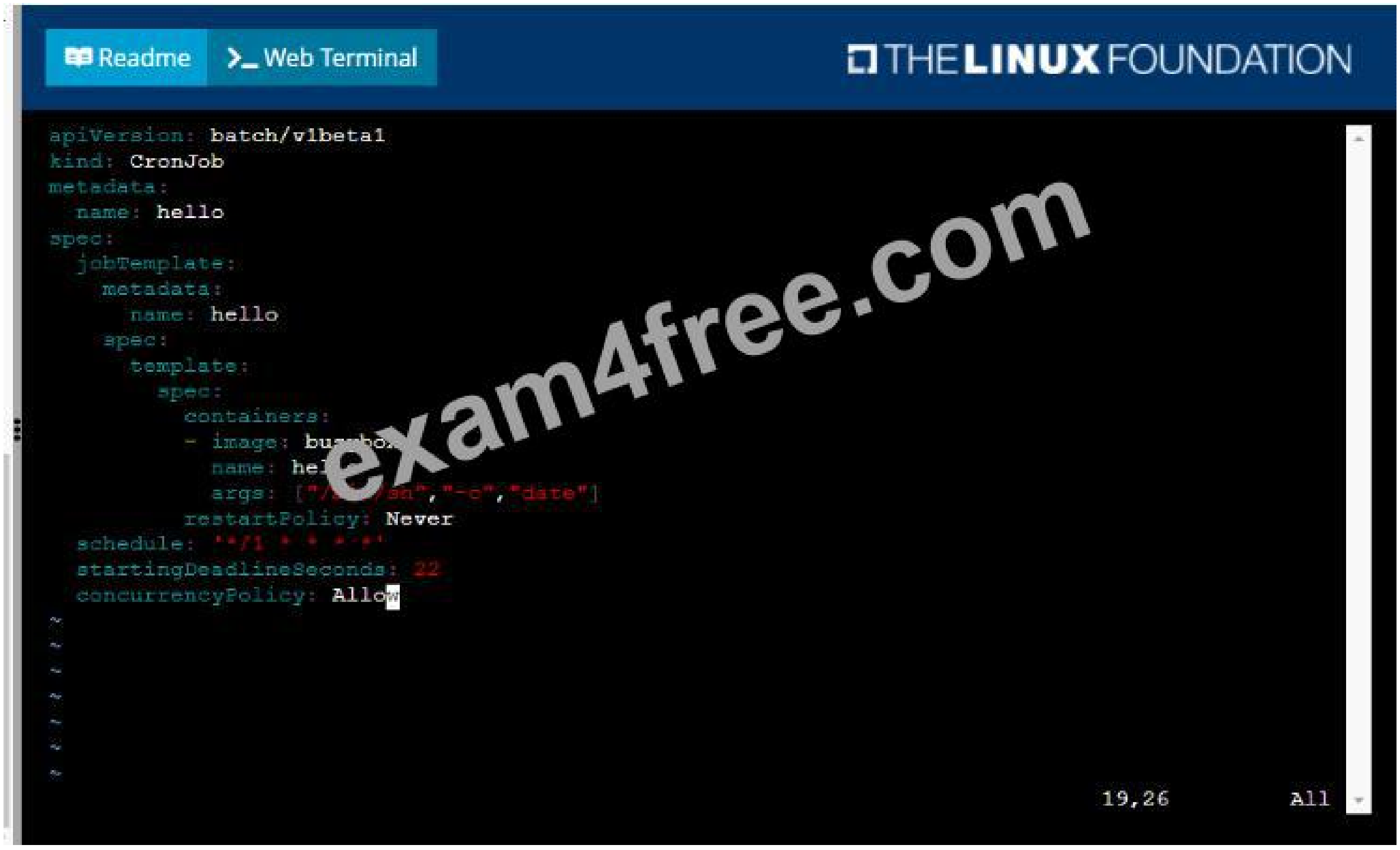


```
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yml > /opt/KDPD00301/periodic.yaml
error: unable to match a printer suitable for the output format "yml", allowed formats are: go-t
emplate,go-template-file,json,jsonpath,jsonpath-as-json,jsonpath-file,name,template,templatefile
,yaml
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yaml > /opt/KDPD00301/periodic.yaml
student@node-1:~$ vim /opt/KDPD00301/periodic.yaml
student@node-1:~$ kubectl create -f /opt/KDPD00301/periodic.yaml
cronjob.batch/hello created
student@node-1:~$ kubectl get cronjob
NAME    SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
hello   */1 * * * *   False     0        <none>          6s
student@node-1:~$
```

B. Solution:



```
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yml > /opt/KDPD00301/periodic.yaml
error: unable to match a printer suitable for the output format "yml", allowed formats are: go-t
emplate,go-template-file,json,jsonpath,jsonpath-as-json,jsonpath-file,name,template,templatefile
,yaml
student@node-1:~$ kubectl create cronjob hello --image=busybox --schedule "* * * * *" --dry-run=
client -o yaml > /opt/KDPD00301/periodic.yaml
student@node-1:~$ vim /opt/KDPD00301/periodic.yaml
```

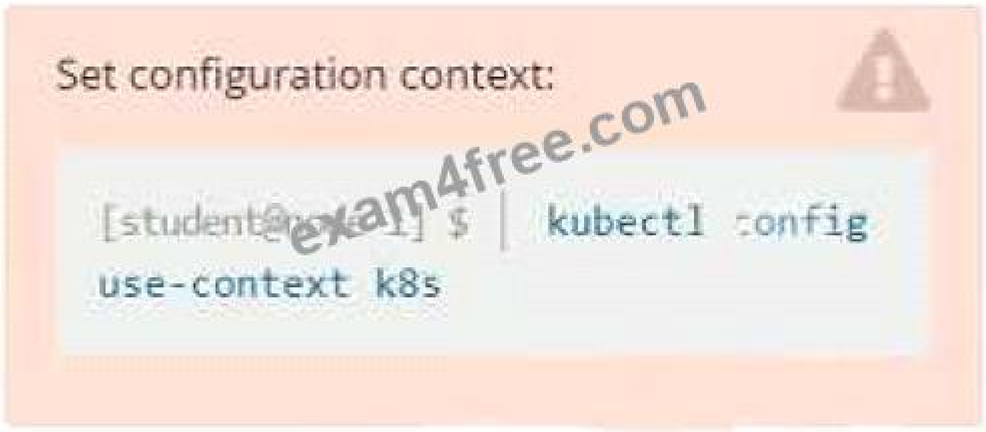


```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: hello
spec:
  jobTemplate:
    metadata:
      name: hello
    spec:
      template:
        spec:
          containers:
          - image: busybox
            name: hello
            args: ["/bin/sh","-c","date"]
          restartPolicy: Never
  schedule: '*/1 * * * *'
  startingDeadlineSeconds: 22
  concurrencyPolicy: Allow
~
~
~
~
~
~
~
                                                        19,26         All
```

Answer: A

NEW QUESTION 20

Context



Set configuration context:

```
[student@node-1] $ | kubectl config
use-context k8s
```

Context
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

Answer:
Explanation:
Solution:

```
apiVersion: v1
kind: Pod
metadata:
labels:
test: liveness
name: liveness-exec
spec:
containers:
- name: liveness
image: k8s.gcr.io/busybox
args:
- /bin/sh
- -c
- touch /tmp/healthy; sleep 30; rm -rf /tmp/healthy; sleep 600
livenessProbe:
exec:
command:
- cat
- /tmp/healthy
initialDelaySeconds: 5
periodSeconds: 5
```

In the configuration file, you can see that the Pod has a single Container. The periodSeconds field specifies that the kubelet should perform a liveness probe every 5 seconds. The initialDelaySeconds field tells the kubelet that it should wait 5 seconds before performing the first probe. To perform a probe, the kubelet executes the command cat /tmp/healthy in the target container. If the command succeeds, it returns 0, and the kubelet considers the container to be alive and healthy. If the command returns a non-zero value, the kubelet kills the container and restarts it.
When the container starts, it executes this command:
/bin/sh -c "touch /tmp/healthy; sleep 30; rm -rf /tmp/healthy; sleep 600" For the first 30 seconds of the container's life, there is a /tmp/healthy file. So during the first 30 seconds, the command cat /tmp/healthy

returns a success code. After 30 seconds, cat /tmp/healthy returns a failure code.
Create the Pod:
kubectl apply -f https://k8s.io/examples/pods/probe/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 "k8s.gcr.io/busybox"
23s 23s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image
"k8s.gcr.io/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
```

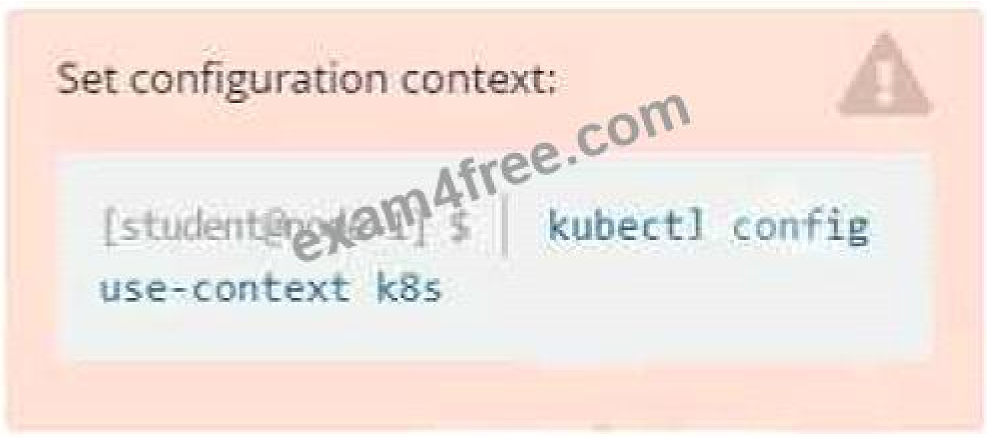
After 35 seconds, view the Pod events again:
kubectl describe pod liveness-exec
At the bottom of the output, there are messages indicating that the liveness probes have failed, and the containers have been killed and recreated.

```
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 "k8s.gcr.io/busybox"
36s 36s 1 {kubelet worker0} spec.containers{liveness} Normal Pulled Successfully pulled image
"k8s.gcr.io/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 1m
```

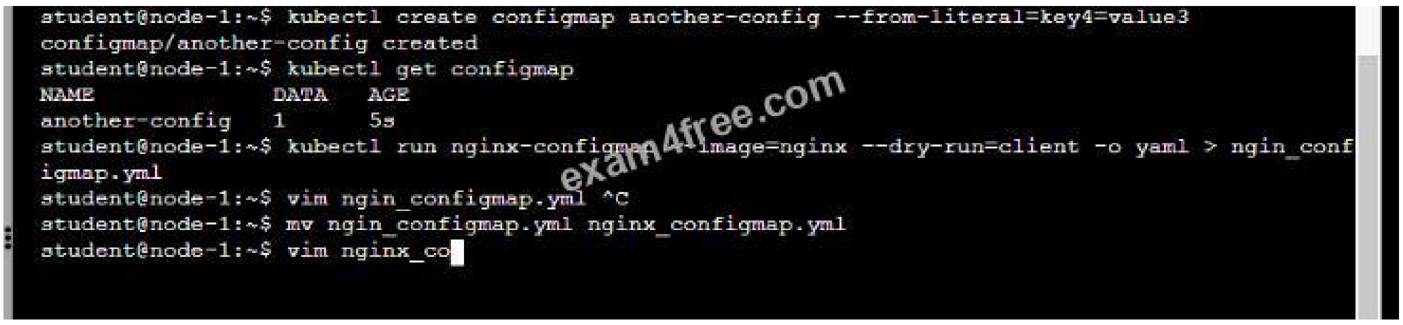
NEW QUESTION 21
Exhibit:



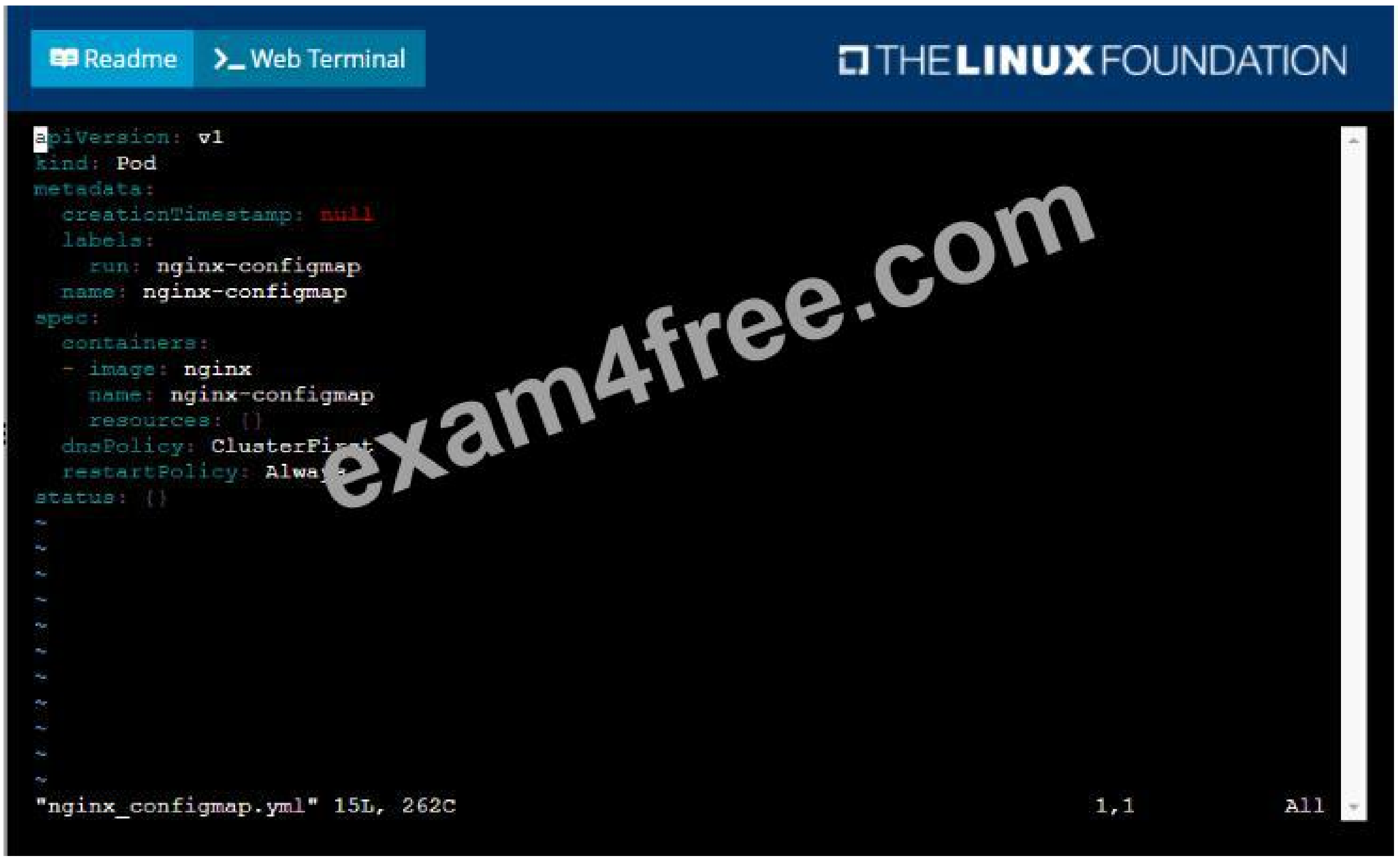
Set configuration context:

```
[student@node-1] $ | kubectl config
use-context k8s
```

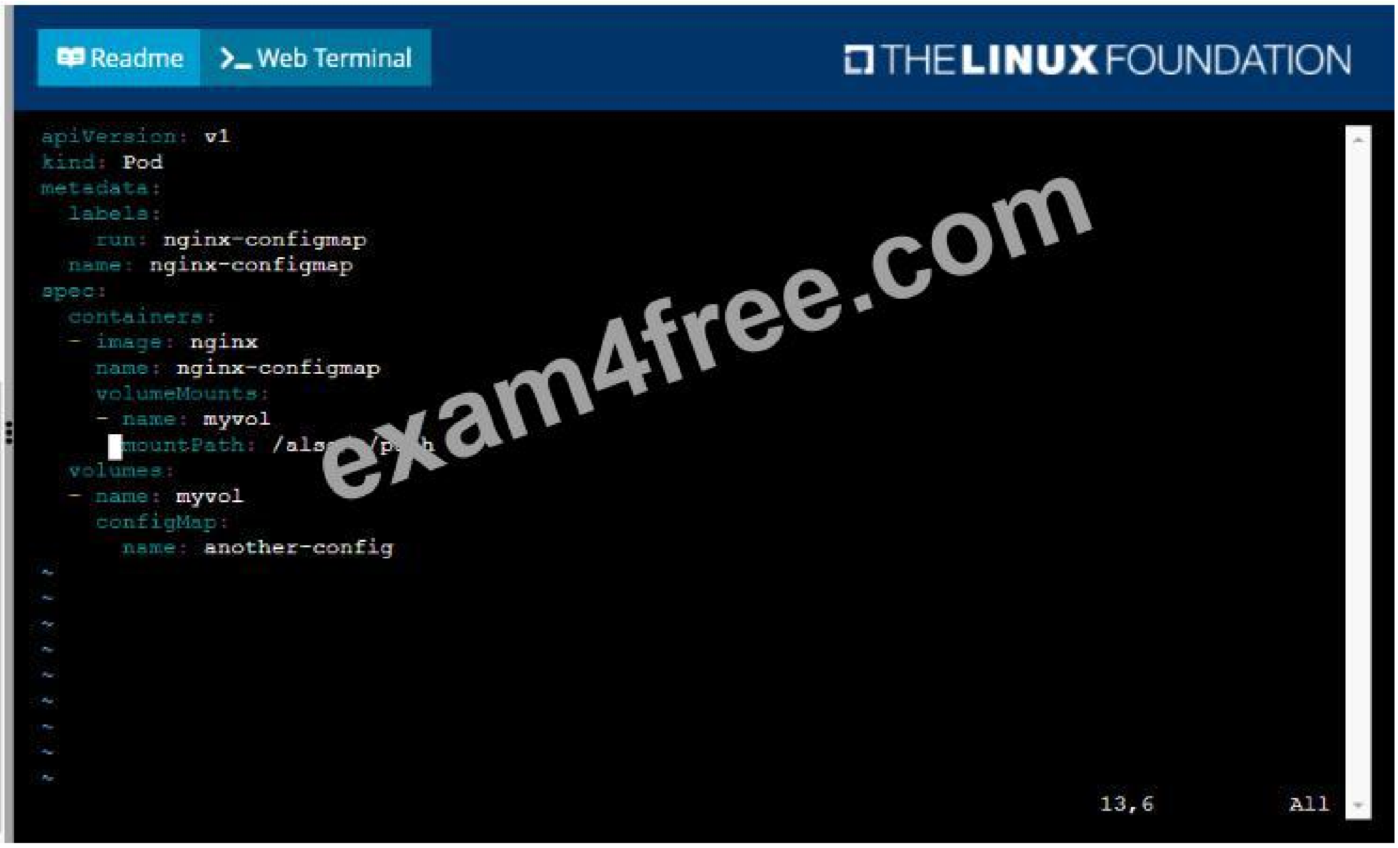
Context
You are tasked to create a ConfigMap and consume the ConfigMap in a pod using a volume mount.
Task
Please complete the following:
* Create a ConfigMap named another-config containing the key/value pair: key4/value3
* start a pod named nginx-configmap containing a single container using the
nginx image, and mount the key you just created into the pod under directory /also/a/path
A. Solution:



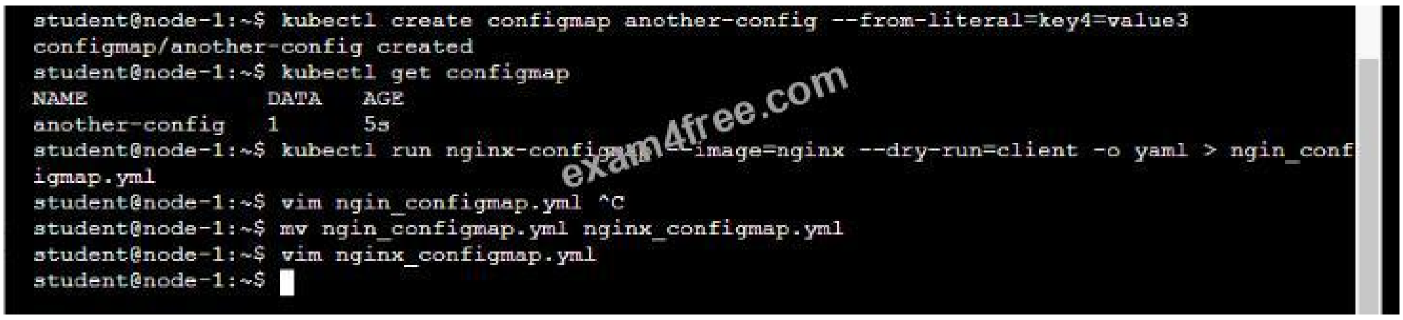
```
student@node-1:~$ kubectl create configmap another-config --from-literal=key4=value3
configmap/another-config created
student@node-1:~$ kubectl get configmap
NAME             DATA   AGE
another-config   1      5s
student@node-1:~$ kubectl run nginx-configmap --image=nginx --dry-run=client -o yaml > ngin_conf
igmap.yml
student@node-1:~$ vim ngin_configmap.yml ^C
student@node-1:~$ mv ngin_configmap.yml nginx_configmap.yml
student@node-1:~$ vim nginx_co
```



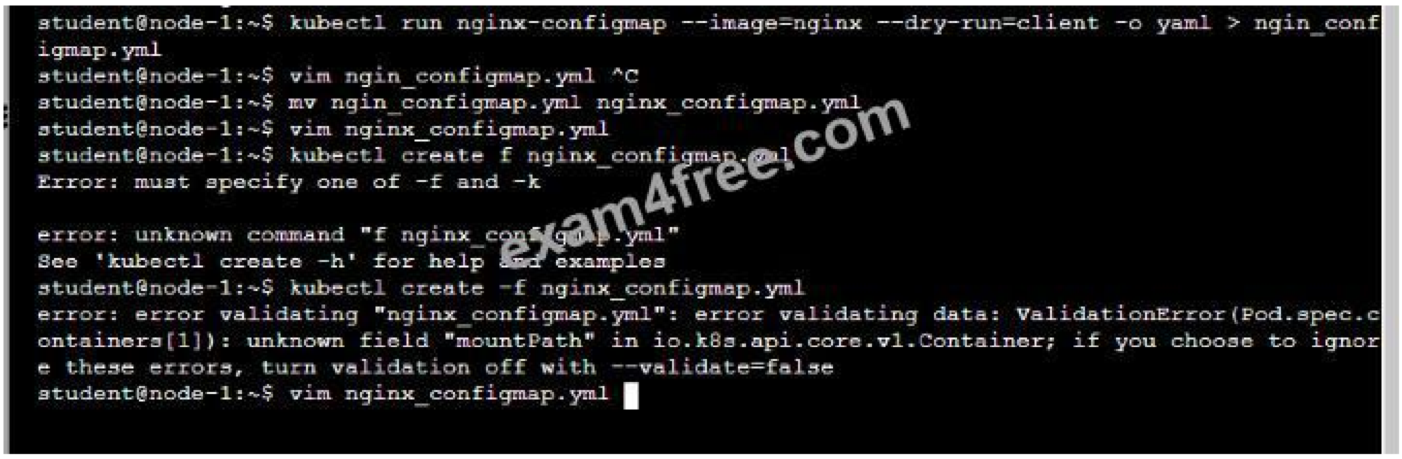
```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-configmap
  name: nginx-configmap
spec:
  containers:
  - image: nginx
    name: nginx-configmap
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
~
~
~
~
~
~
~
~
~
~
~
"nginx_configmap.yml" 15L, 262C                                          1,1           All
```



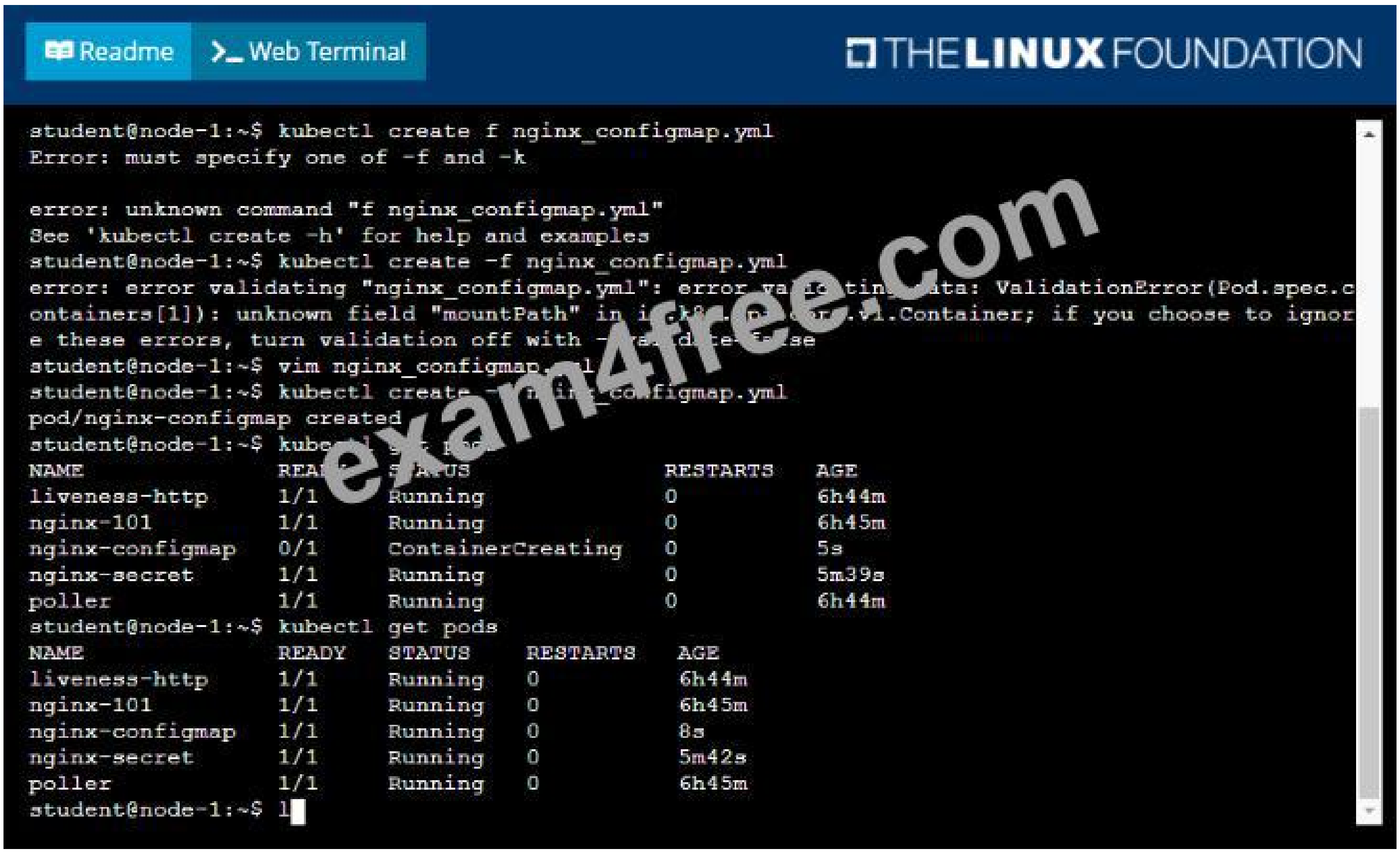
```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx-configmap
  name: nginx-configmap
spec:
  containers:
  - image: nginx
    name: nginx-configmap
    volumeMounts:
    - name: myvol
      mountPath: /also/path
  volumes:
  - name: myvol
    configMap:
      name: another-config
~
~
~
~
~
~
~
~
~
                                                                         13,6          All
```



```
student@node-1:~$ kubectl create configmap another-config --from-literal=key4=value3
configmap/another-config created
student@node-1:~$ kubectl get configmap
NAME             DATA   AGE
another-config   1      5s
student@node-1:~$ kubectl run nginx-configmap --image=nginx --dry-run=client -o yaml > ngin_conf
igmap.yml
student@node-1:~$ vim ngin_configmap.yml ^C
student@node-1:~$ mv ngin_configmap.yml nginx_configmap.yml
student@node-1:~$ vim nginx_configmap.yml
student@node-1:~$
```

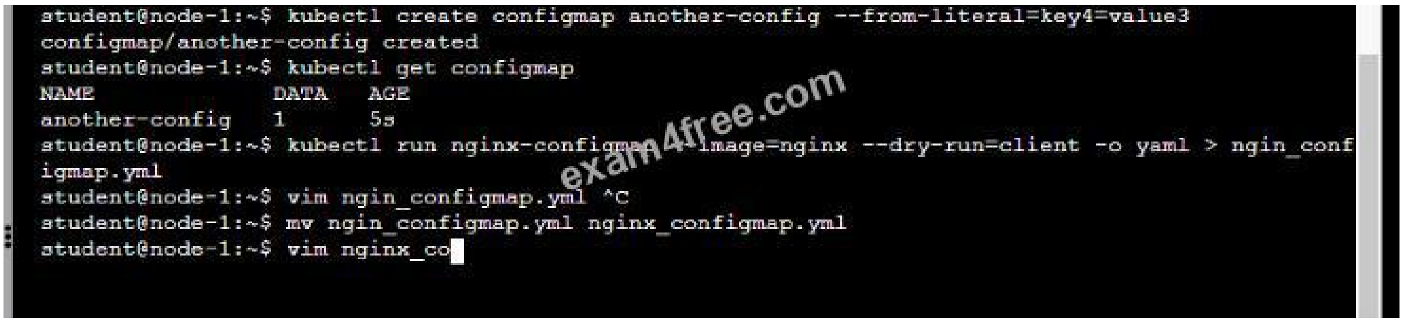


```
student@node-1:~$ kubectl run nginx-configmap --image=nginx --dry-run=client -o yaml > ngin_conf
igmap.yml
student@node-1:~$ vim ngin_configmap.yml ^C
student@node-1:~$ mv ngin_configmap.yml nginx_configmap.yml
student@node-1:~$ vim nginx_configmap.yml
student@node-1:~$ kubectl create f nginx_configmap.yml
Error: must specify one of -f and -k

error: unknown command "f nginx_configmap.yml"
See 'kubectl create -h' for help and examples
student@node-1:~$ kubectl create -f nginx_configmap.yml
error: error validating "nginx_configmap.yml": error validating data: ValidationError(Pod.spec.c
ontainers[1]): unknown field "mountPath" in io.k8s.api.core.v1.Container; if you choose to ignor
e these errors, turn validation off with --validate=false
student@node-1:~$ vim nginx_configmap.yml
```

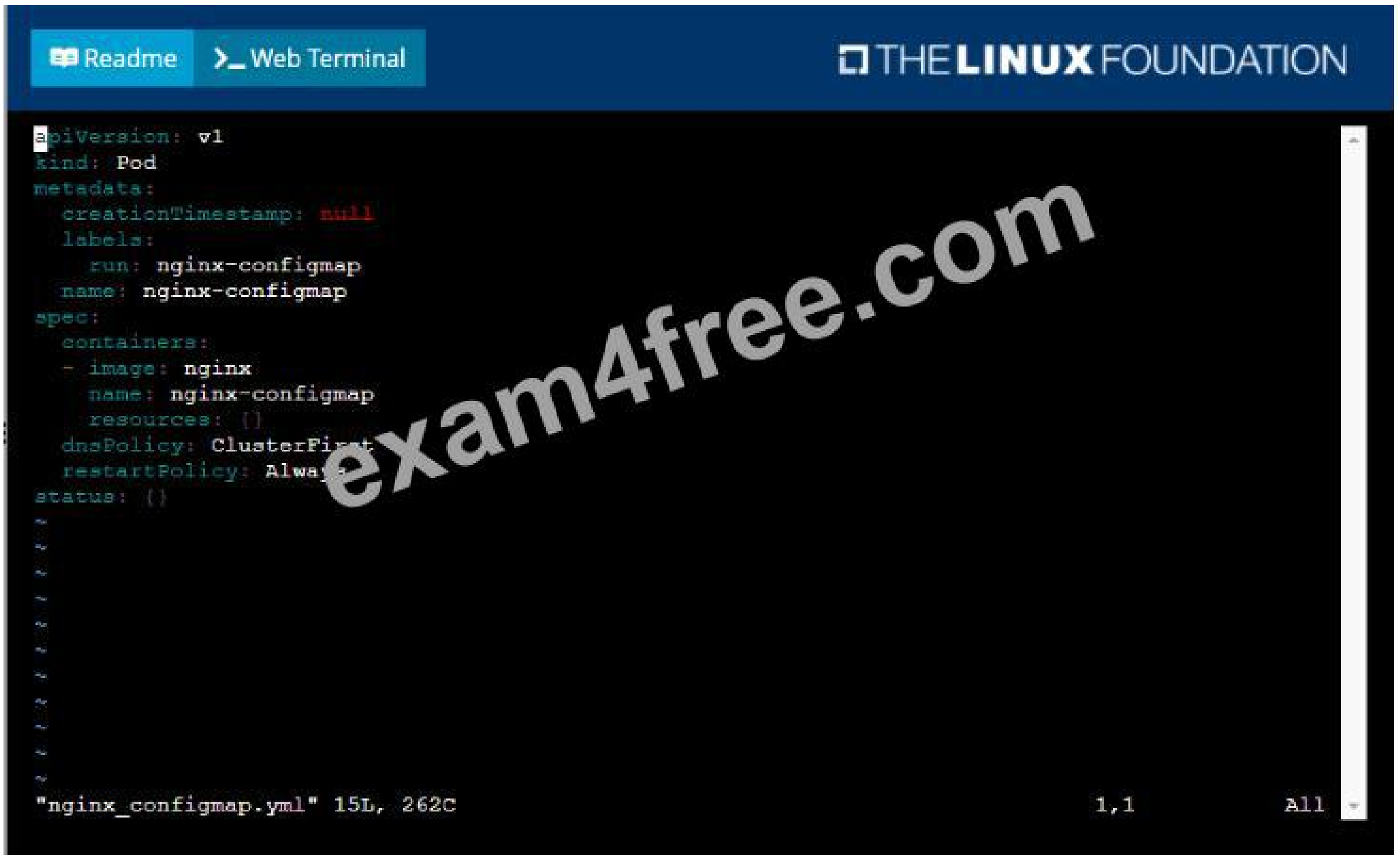


```
student@node-1:~$ kubectl create f nginx_configmap.yml
Error: must specify one of -f and -k

error: unknown command "f nginx_configmap.yml"
See 'kubectl create -h' for help and examples
student@node-1:~$ kubectl create -f nginx_configmap.yml
error: error validating "nginx_configmap.yml": error validating data: ValidationError(Pod.spec.c
ontainers[1]): unknown field "mountPath" in io.k8s.api.core.v1.Container; if you choose to ignor
e these errors, turn validation off with --validate=false
student@node-1:~$ vim nginx_configmap.yml
student@node-1:~$ kubectl create -f nginx_configmap.yml
pod/nginx-configmap created
student@node-1:~$ kubectl get pods
NAME              READY   STATUS              RESTARTS   AGE
liveness-http     1/1     Running             0          6h44m
nginx-101         1/1     Running             0          6h45m
nginx-configmap   0/1     ContainerCreating   0          5s
nginx-secret      1/1     Running             0          5m39s
poller            1/1     Running             0          6h44m
student@node-1:~$ kubectl get pods
NAME              READY   STATUS    RESTARTS   AGE
liveness-http     1/1     Running   0          6h44m
nginx-101         1/1     Running   0          6h45m
nginx-configmap   1/1     Running   0          8s
nginx-secret      1/1     Running   0          5m42s
poller            1/1     Running   0          6h45m
student@node-1:~$ l
```

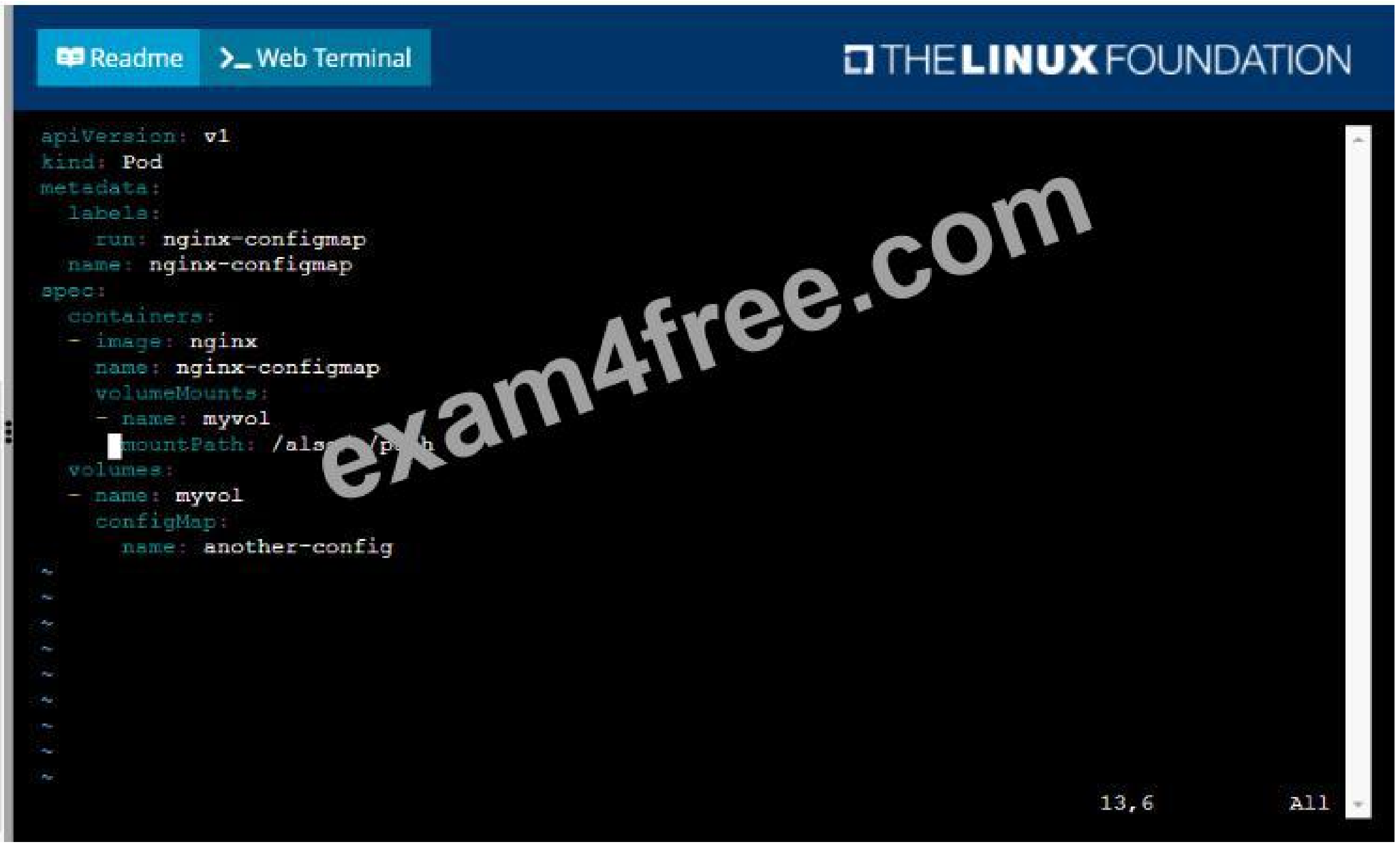
B. Solution:



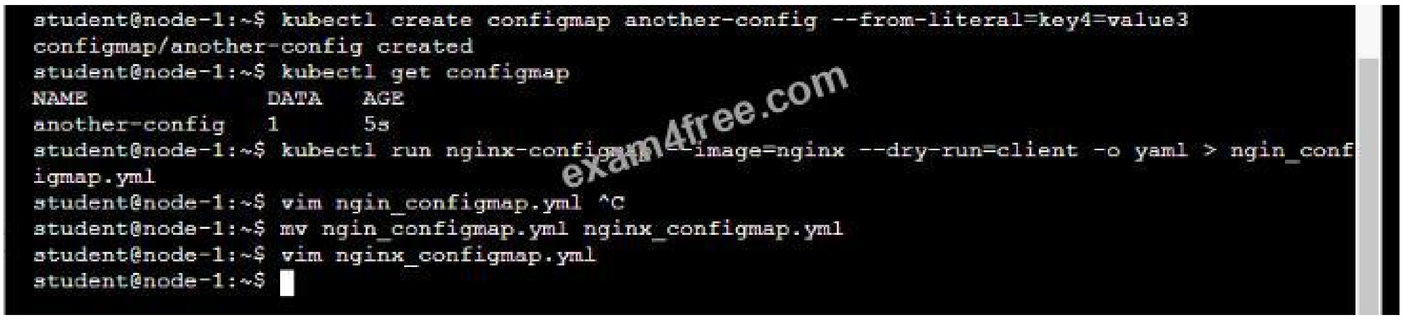
```
student@node-1:~$ kubectl create configmap another-config --from-literal=key4=value3
configmap/another-config created
student@node-1:~$ kubectl get configmap
NAME             DATA   AGE
another-config   1      5s
student@node-1:~$ kubectl run nginx-configmap --image=nginx --dry-run=client -o yaml > ngin_conf
igmap.yml
student@node-1:~$ vim ngin_configmap.yml ^C
student@node-1:~$ mv ngin_configmap.yml nginx_configmap.yml
student@node-1:~$ vim nginx_co
```



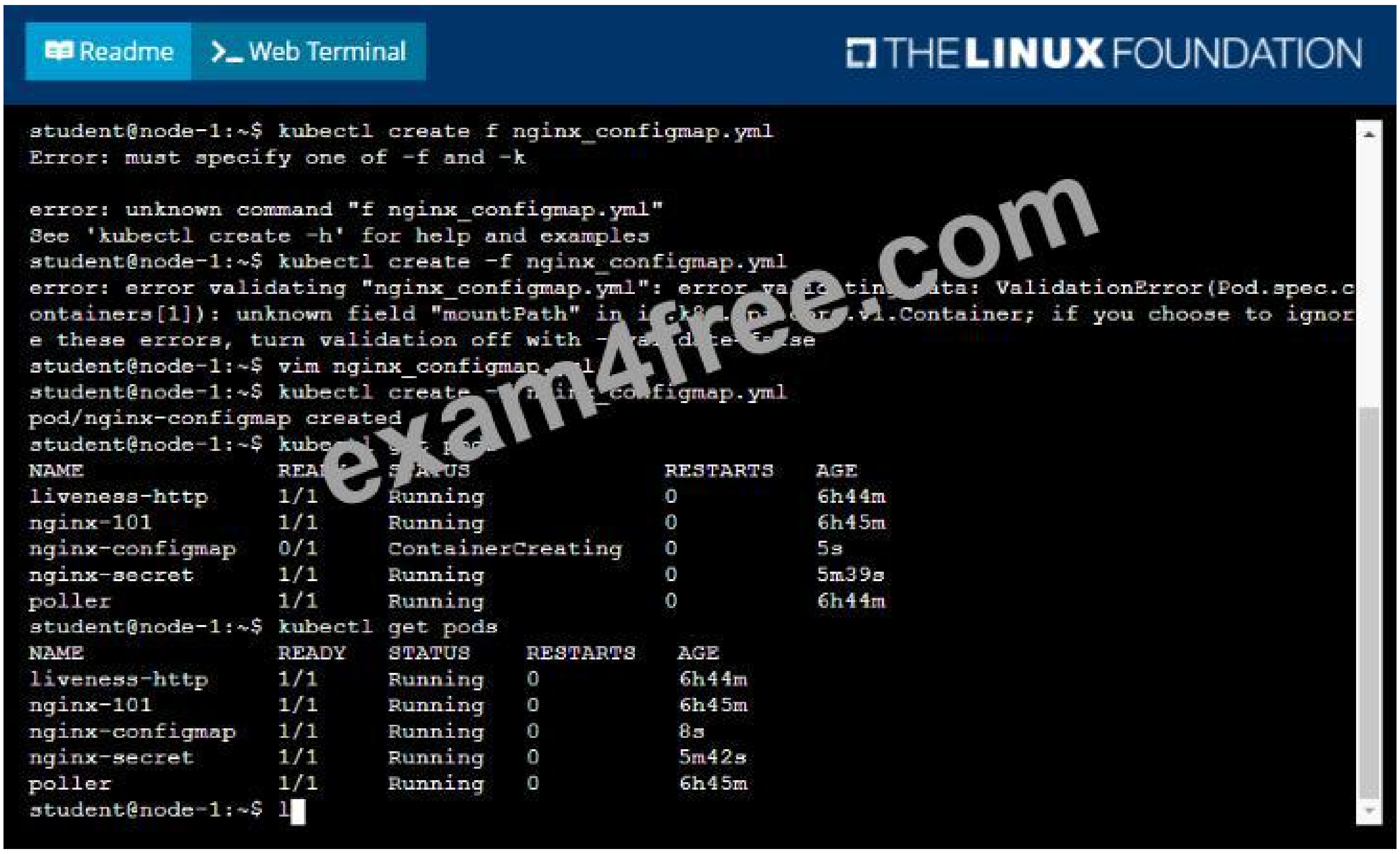
```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-configmap
  name: nginx-configmap
spec:
  containers:
  - image: nginx
    name: nginx-configmap
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
~
~
~
~
~
~
~
~
~
~
~
"nginx_configmap.yml" 15L, 262C                                         1,1           All
```



```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx-configmap
  name: nginx-configmap
spec:
  containers:
  - image: nginx
    name: nginx-configmap
    volumeMounts:
    - name: myvol
      mountPath: /als[illegible]/p[illegible]h
  volumes:
  - name: myvol
    configMap:
      name: another-config
~
~
~
~
~
~
~
~
~
                                                                  13,6          All
```



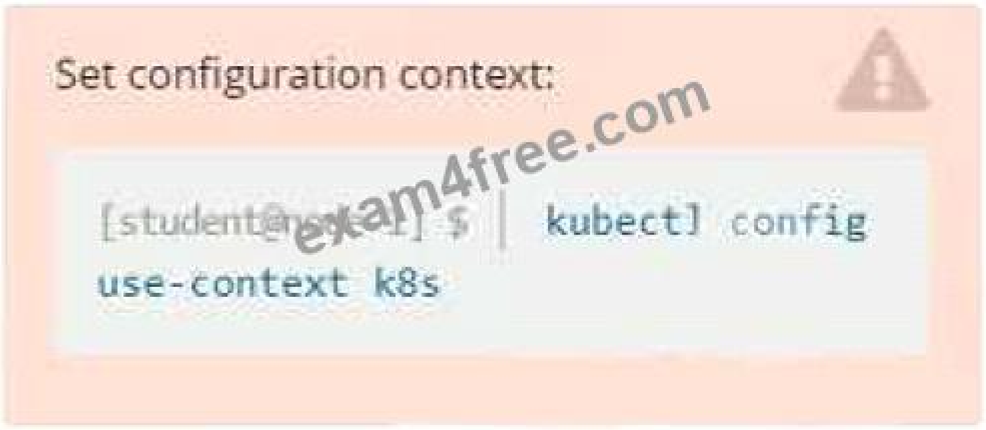
```
student@node-1:~$ kubectl create configmap another-config --from-literal=key4=value3
configmap/another-config created
student@node-1:~$ kubectl get configmap
NAME              DATA   AGE
another-config    1      5s
student@node-1:~$ kubectl run nginx-configmap --image=nginx --dry-run=client -o yaml > ngin_conf
igmap.yml
student@node-1:~$ vim ngin_configmap.yml ^C
student@node-1:~$ mv ngin_configmap.yml nginx_configmap.yml
student@node-1:~$ vim nginx_configmap.yml
student@node-1:~$ 
```



```
student@node-1:~$ kubectl create f nginx_configmap.yml
Error: must specify one of -f and -k

error: unknown command "f nginx_configmap.yml"
See 'kubectl create -h' for help and examples
student@node-1:~$ kubectl create -f nginx_configmap.yml
error: error validating "nginx_configmap.yml": error validating data: ValidationError(Pod.spec.c
ontainers[1]): unknown field "mountPath" in io.k8s.api.core.v1.Container; if you choose to ignor
e these errors, turn validation off with --validate=false
student@node-1:~$ vim nginx_configmap.yml
student@node-1:~$ kubectl create -f nginx_configmap.yml
pod/nginx-configmap created
student@node-1:~$ kubectl get pods
NAME              READY   STATUS              RESTARTS   AGE
liveness-http     1/1     Running             0          6h44m
nginx-101         1/1     Running             0          6h45m
nginx-configmap   0/1     ContainerCreating   0          5s
nginx-secret      1/1     Running             0          5m39s
poller            1/1     Running             0          6h44m
student@node-1:~$ kubectl get pods
NAME              READY   STATUS    RESTARTS   AGE
liveness-http     1/1     Running   0          6h44m
nginx-101         1/1     Running   0          6h45m
nginx-configmap   1/1     Running   0          8s
nginx-secret      1/1     Running   0          5m42s
poller            1/1     Running   0          6h45m
student@node-1:~$ l
```

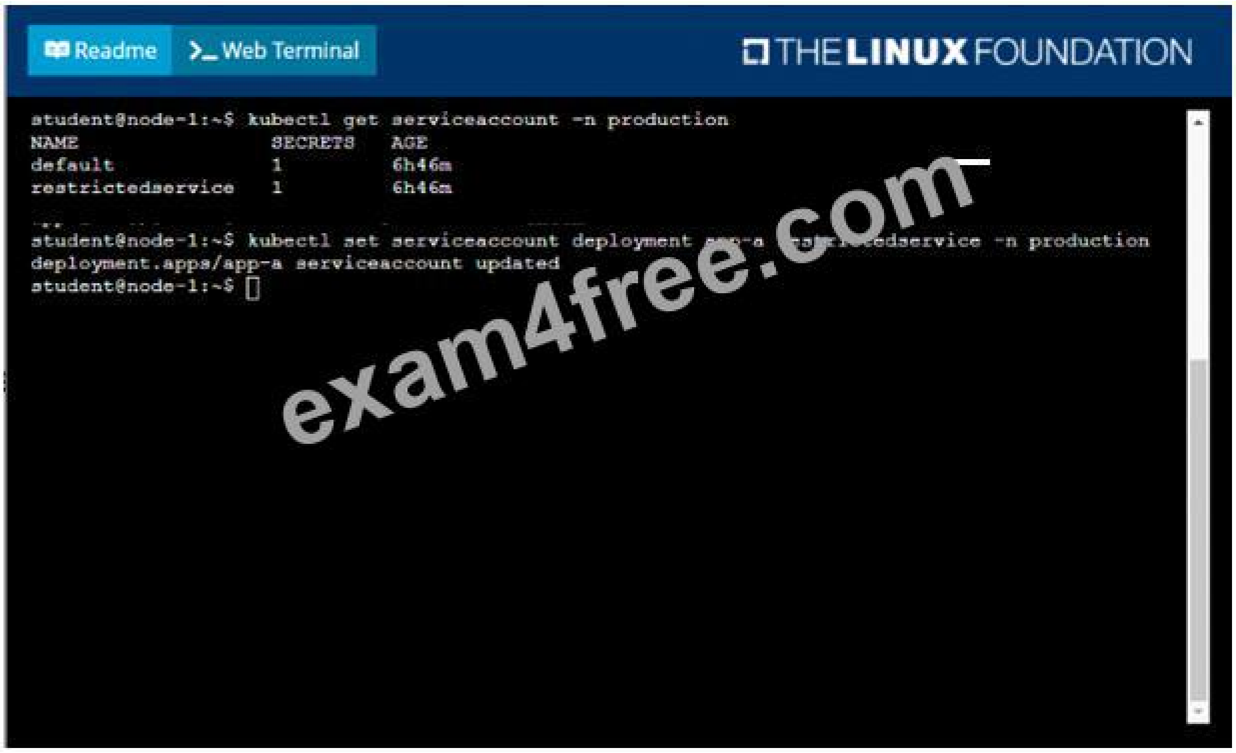
Answer: A

NEW QUESTION 22
Exhibit:


Set configuration context:

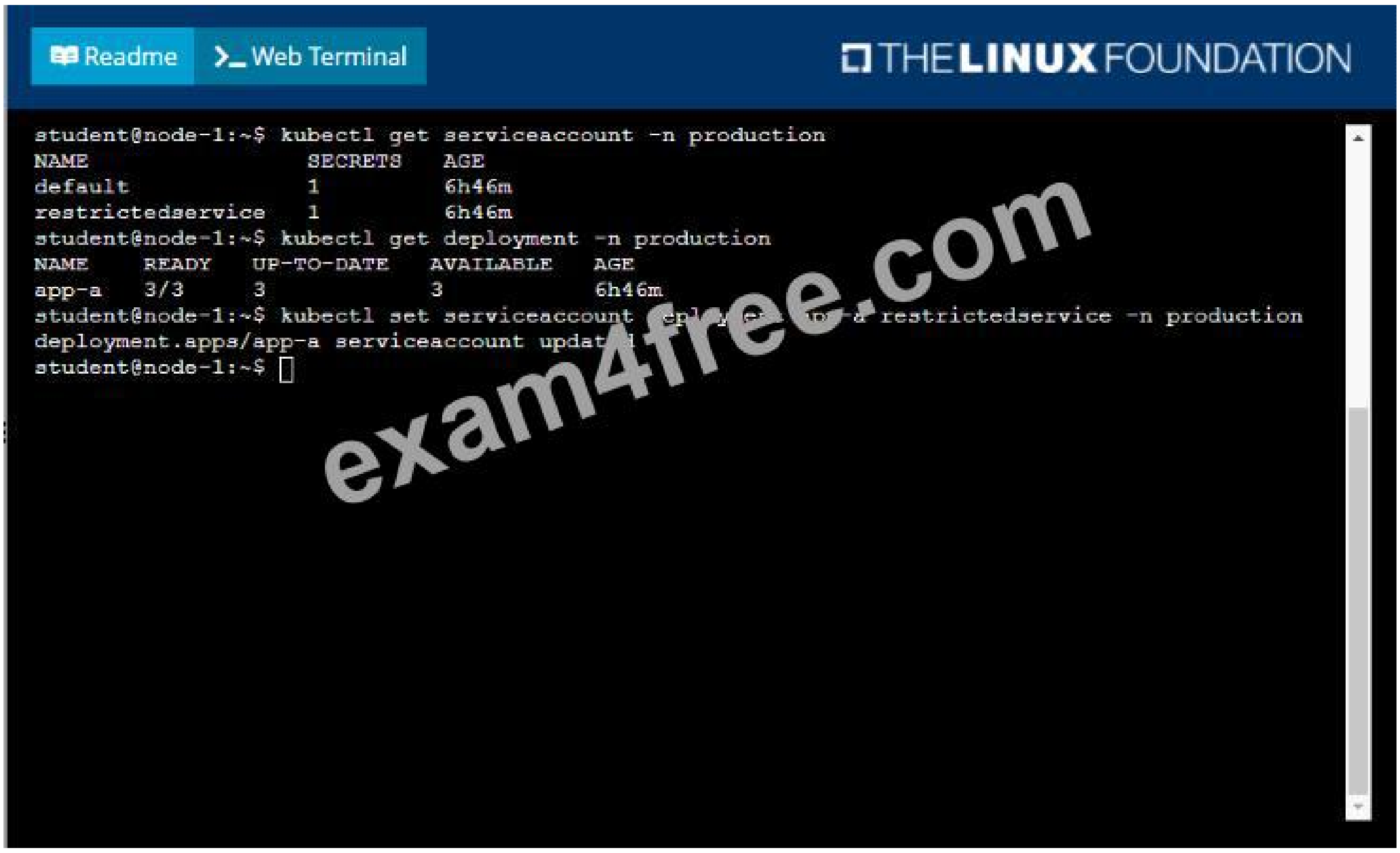
```
[student@node-1] $ | kubectl config
use-context k8s
```

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.
A. Solution:



```
student@node-1:~$ kubectl get serviceaccount -n production
NAME                SECRETS    AGE
default             1          6h46m
restrictedservice   1          6h46m

student@node-1:~$ kubectl set serviceaccount deployment app-a restrictedservice -n production
deployment.apps/app-a serviceaccount updated
student@node-1:~$
```

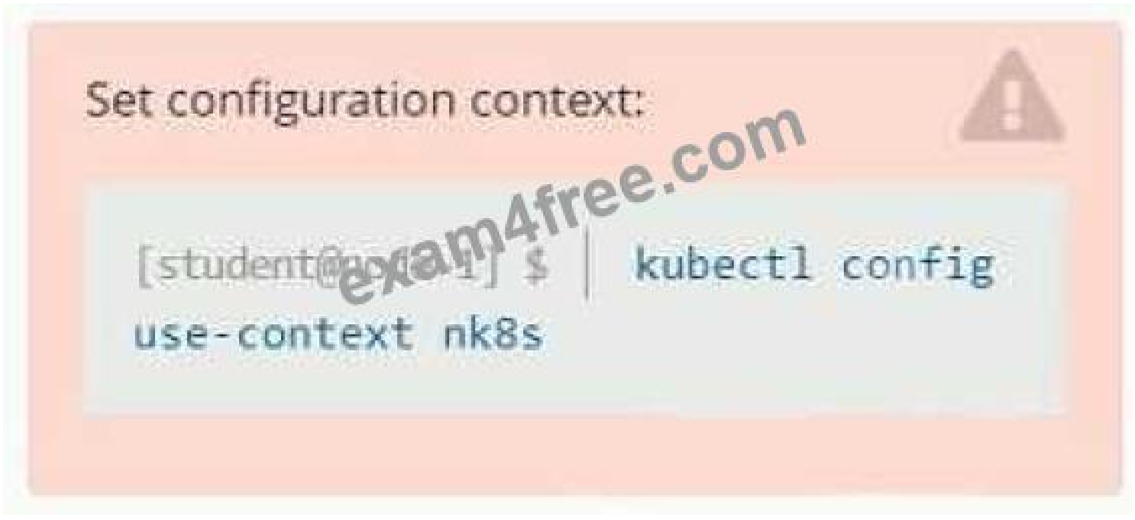
B. Solution:



```
student@node-1:~$ kubectl get serviceaccount -n production
NAME                SECRETS    AGE
default             1          6h46m
restrictedservice   1          6h46m
student@node-1:~$ kubectl get deployment -n production
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
app-a   3/3     3            3           6h46m
student@node-1:~$ kubectl set serviceaccount deployment app-a restrictedservice -n production
deployment.apps/app-a serviceaccount updated
student@node-1:~$
```

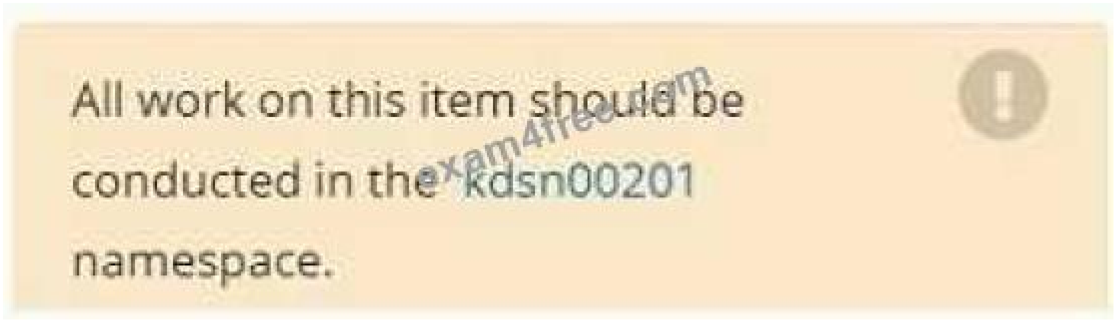
Answer: B

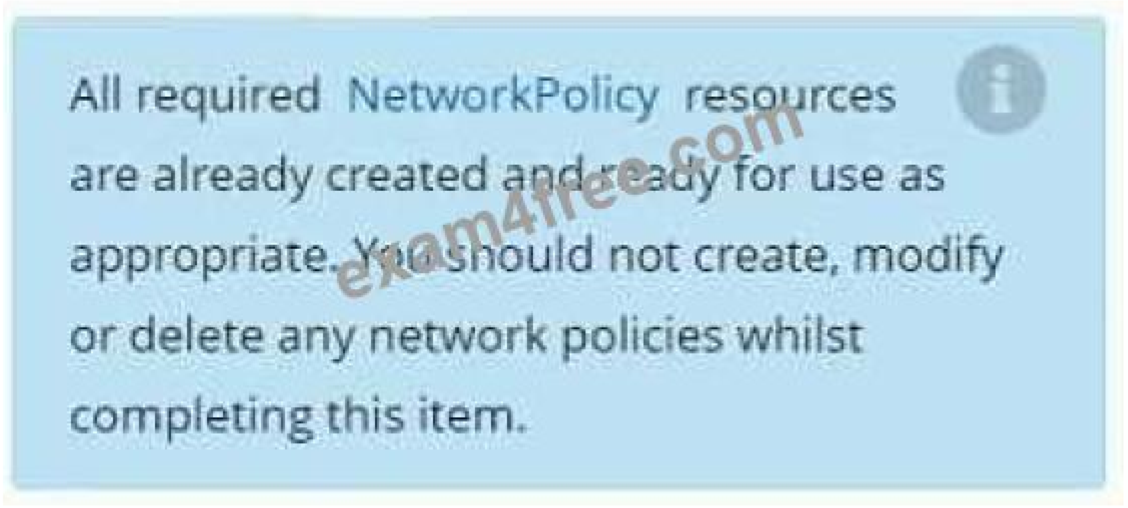
NEW QUESTION 23


Set configuration context:

```
[student@node-1] $ | kubectl config
use-context nk8s
```

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.

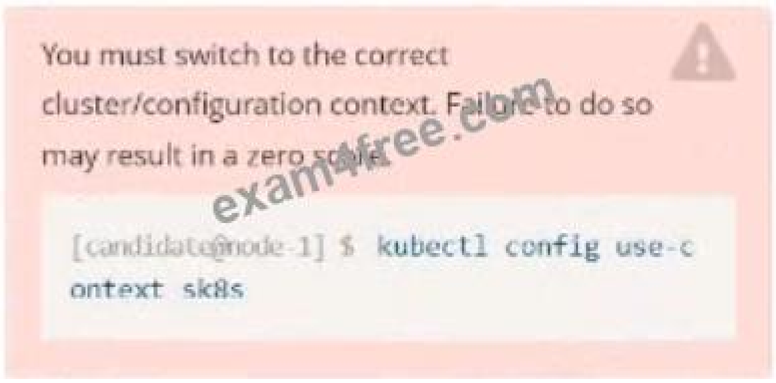

All work on this item should be conducted in the kdsn00201 namespace.


All required NetworkPolicy resources are already created and ready for use as appropriate. You should not create, modify or delete any network policies whilst completing this item.

Answer:
Explanation:
See the solution below.
Explanation
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 24


You must switch to the correct cluster/configuration context. Failure to do so may result in a zero score.

```
[candidate@node-1] $ kubectl config use-c
ontext sk8s
```

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

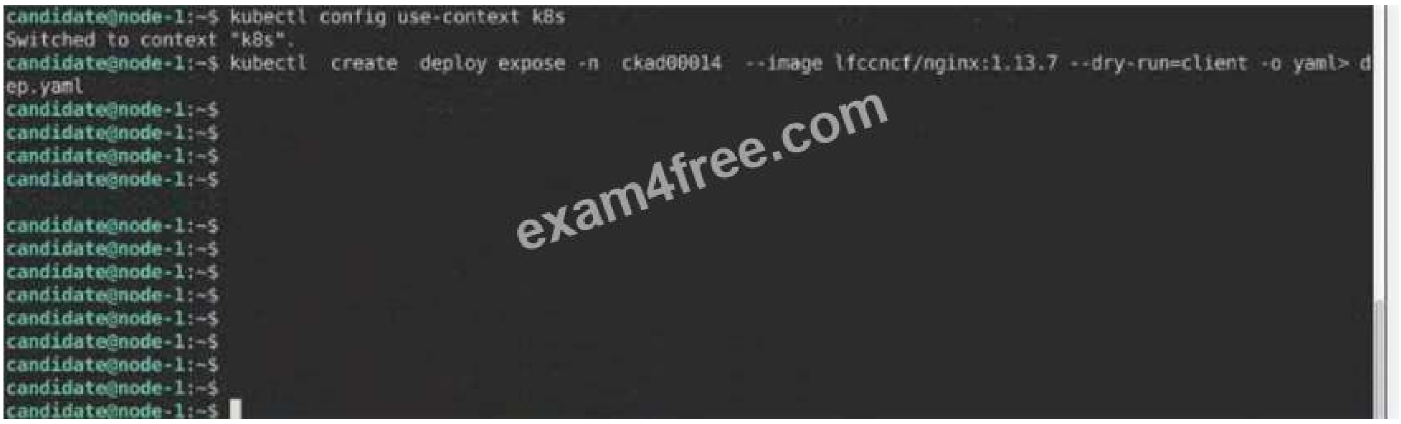
Answer:

Explanation:

See the solution below.

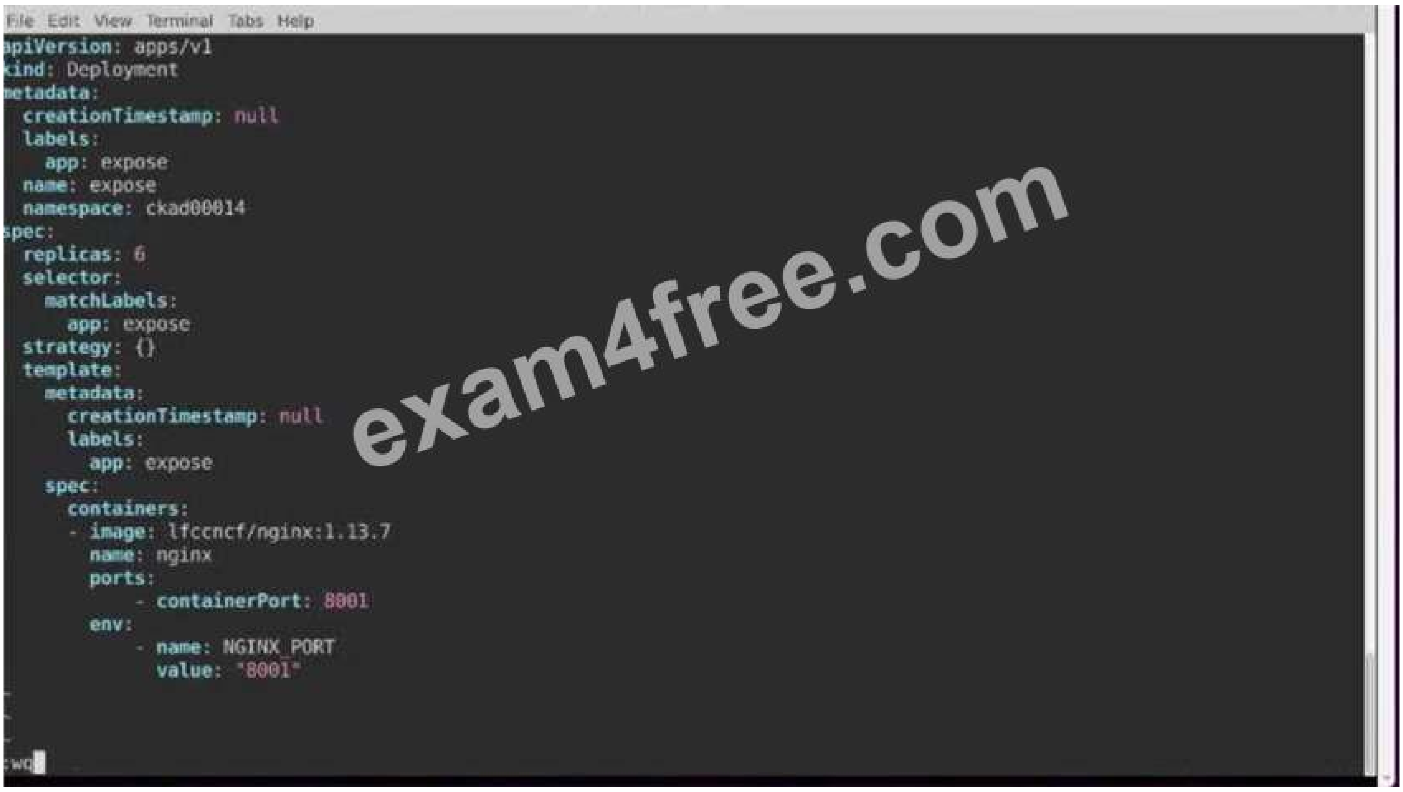
Explanation

Solution:



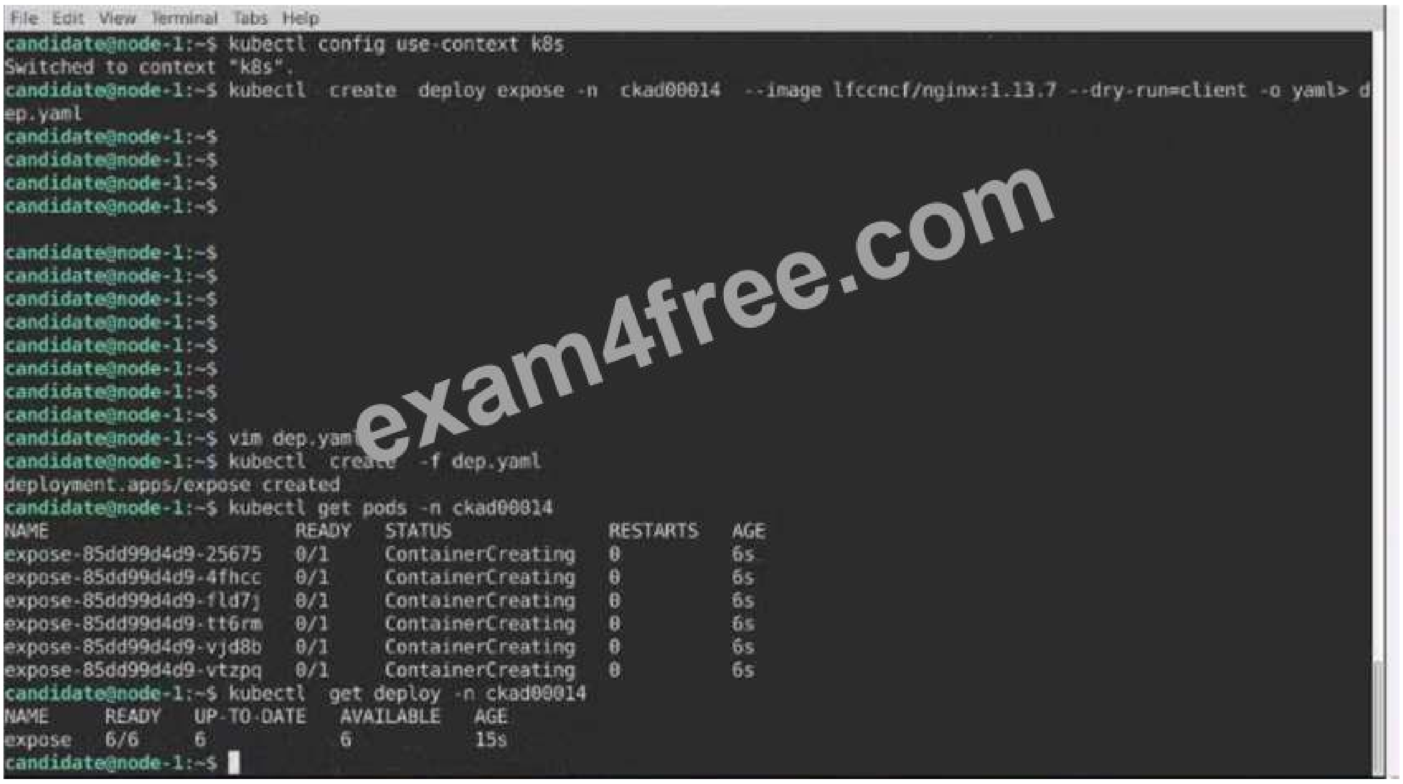
```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl  create  deploy expose -n  ckad00014  --image lfccncf/nginx:1.13.7 --dry-run=client -o yaml> d
ep.yaml
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$

candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
```

Text Description automatically generated



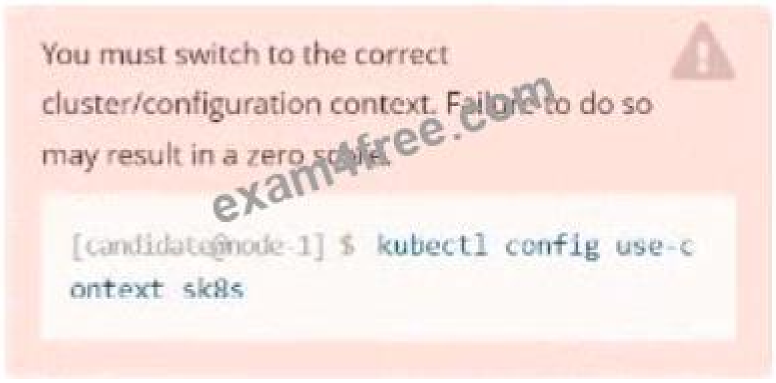
```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: expose
  name: expose
  namespace: ckad00014
spec:
  replicas: 6
  selector:
    matchLabels:
      app: expose
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: expose
    spec:
      containers:
      - image: lfccncf/nginx:1.13.7
        name: nginx
        ports:
           - containerPort: 8001
        env:
           - name: NGINX_PORT
             value: "8001"
~
~
:wq
```

Text Description automatically generated



```
File  Edit  View  Terminal  Tabs  Help
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl  create  deploy expose -n  ckad00014  --image lfccncf/nginx:1.13.7 --dry-run=client -o yaml> d
ep.yaml
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$

candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$
candidate@node-1:~$ vim dep.yaml
candidate@node-1:~$ kubectl  create  -f dep.yaml
deployment.apps/expose created
candidate@node-1:~$ kubectl get pods -n ckad00014
NAME                       READY   STATUS              RESTARTS   AGE
expose-85dd99d4d9-25675    0/1     ContainerCreating   0          6s
expose-85dd99d4d9-4fhcc    0/1     ContainerCreating   0          6s
expose-85dd99d4d9-fld7j    0/1     ContainerCreating   0          6s
expose-85dd99d4d9-tt6rm    0/1     ContainerCreating   0          6s
expose-85dd99d4d9-vjd8b    0/1     ContainerCreating   0          6s
expose-85dd99d4d9-vtzpq    0/1     ContainerCreating   0          6s
candidate@node-1:~$ kubectl  get deploy -n ckad00014
NAME     READY   UP-TO-DATE   AVAILABLE   AGE
expose   6/6     6            6           15s
candidate@node-1:~$
```

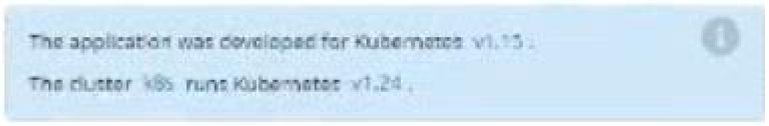
NEW QUESTION 25
Context



You must switch to the correct cluster/configuration context. Failure to do so may result in a zero score.

```
[candidate@node-1] $ kubectl config use-context sk8s
```

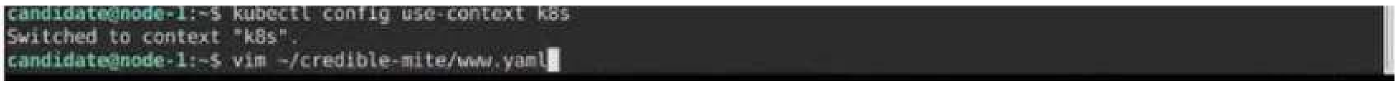
Task:
1) Fix any API depreciation issues in the manifest file -/credible-mite/www.yaml so that this application can be deployed on cluster K8s.



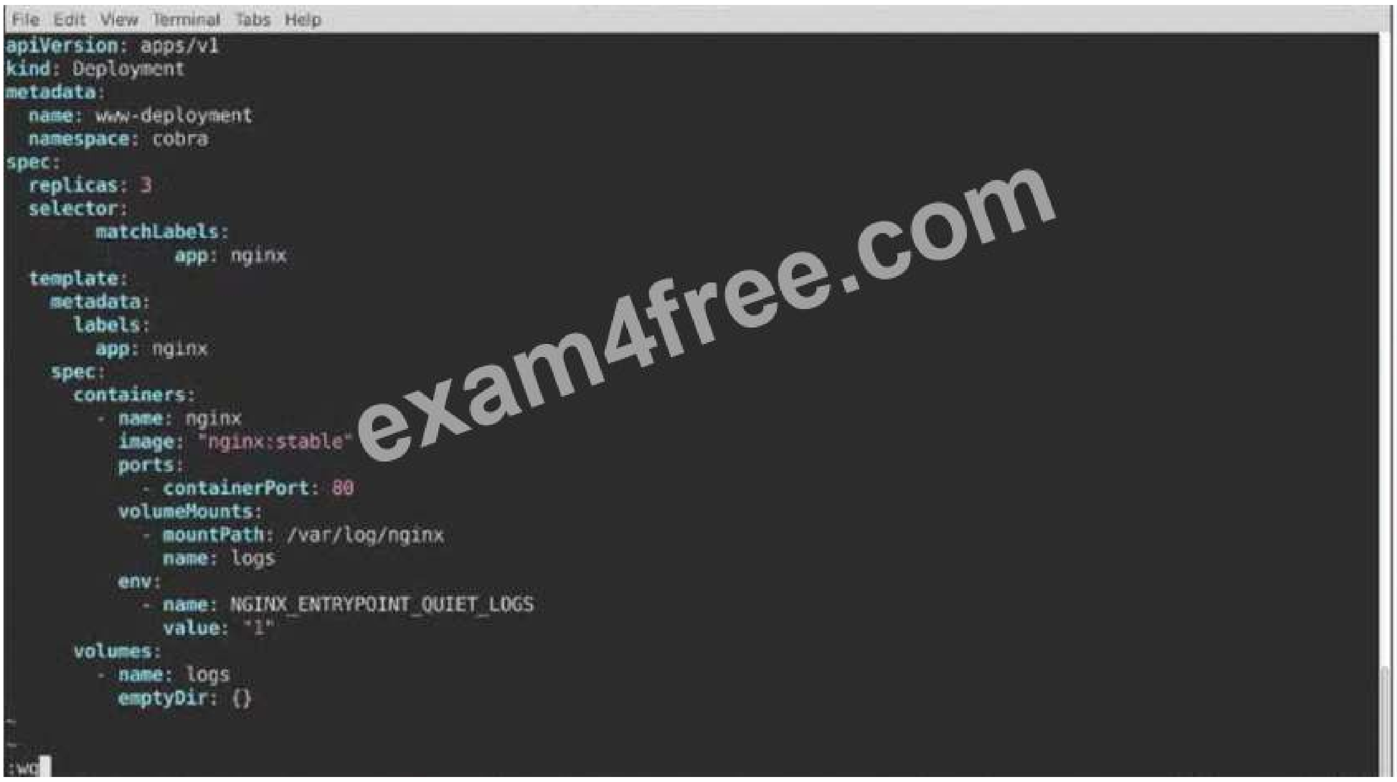
The application was developed for Kubernetes v1.15 .
The cluster k8s runs Kubernetes v1.24 .

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

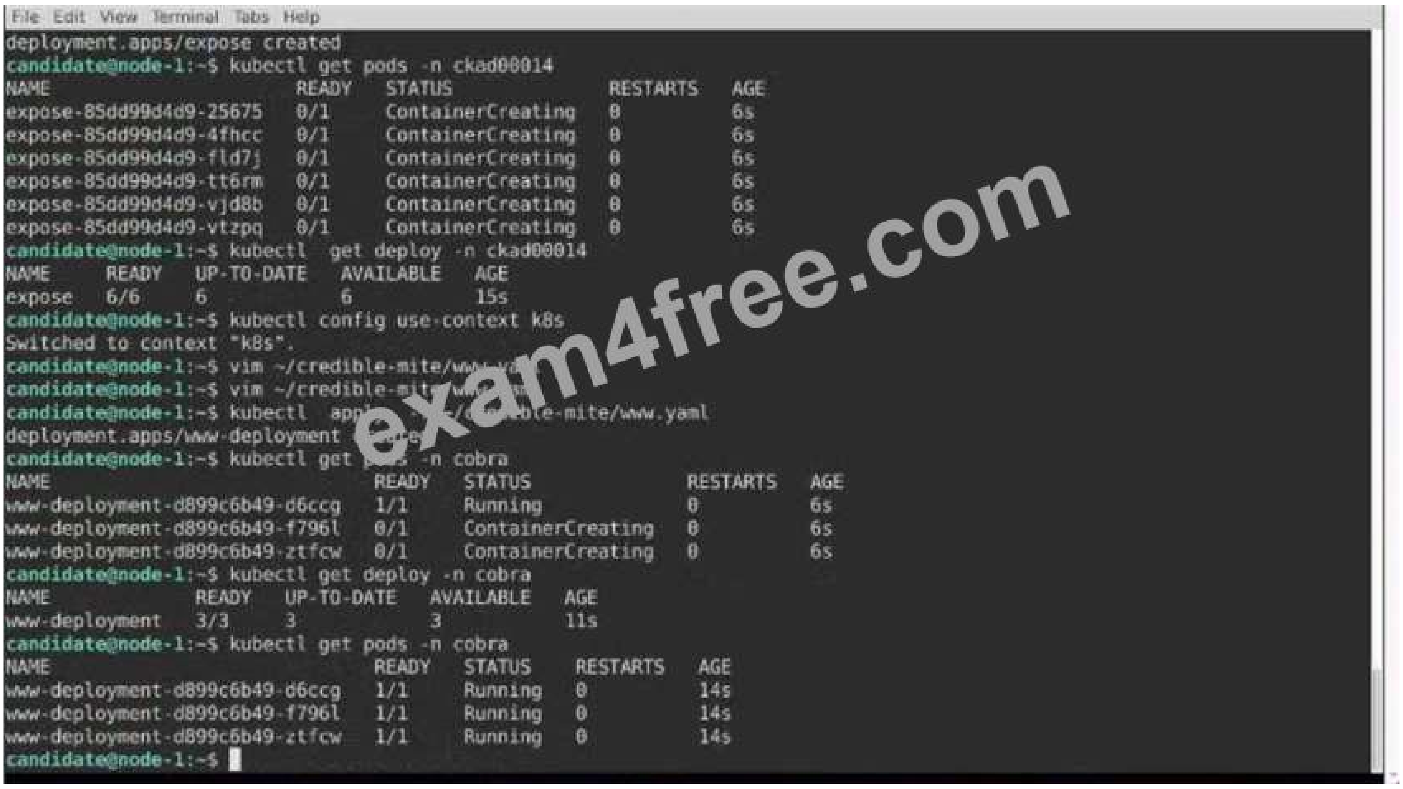
Answer:
Explanation:
Solution:



```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/credible-mite/www.yaml
```

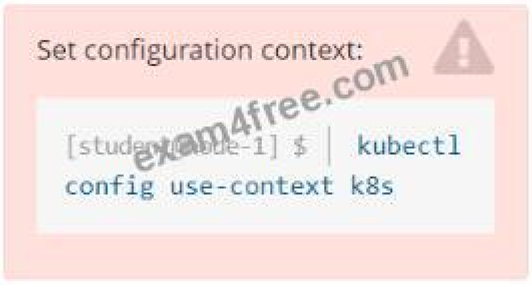


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: www-deployment
  namespace: cobra
spec:
  replicas: 3
  selector:
        matchLabels:
               app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: "nginx:stable"
        ports:
        - containerPort: 80
        volumeMounts:
        - mountPath: /var/log/nginx
          name: logs
        env:
        - name: NGINX_ENTRYPOINT_QUIET_LOGS
          value: "1"
      volumes:
      - name: logs
        emptyDir: {}
~
~
:wq
```



```
deployment.apps/expose created
candidate@node-1:~$ kubectl get pods -n ckad00014
NAME                        READY   STATUS              RESTARTS   AGE
expose-85dd99d4d9-25675     0/1     ContainerCreating   0          6s
expose-85dd99d4d9-4fhcc     0/1     ContainerCreating   0          6s
expose-85dd99d4d9-fld7j     0/1     ContainerCreating   0          6s
expose-85dd99d4d9-tt6rm     0/1     ContainerCreating   0          6s
expose-85dd99d4d9-vjd8b     0/1     ContainerCreating   0          6s
expose-85dd99d4d9-vtzpq     0/1     ContainerCreating   0          6s
candidate@node-1:~$ kubectl  get deploy -n ckad00014
NAME     READY   UP-TO-DATE   AVAILABLE   AGE
expose   6/6     6            6           15s
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/credible-mite/www.yaml
candidate@node-1:~$ vim ~/credible-mite/www.yaml
candidate@node-1:~$ kubectl  apply -f ~/credible-mite/www.yaml
deployment.apps/www-deployment created
candidate@node-1:~$ kubectl get pods -n cobra
NAME                              READY   STATUS              RESTARTS   AGE
www-deployment-d899c6b49-d6ccg    1/1     Running             0          6s
www-deployment-d899c6b49-f796l    0/1     ContainerCreating   0          6s
www-deployment-d899c6b49-ztfcw    0/1     ContainerCreating   0          6s
candidate@node-1:~$ kubectl get deploy -n cobra
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
www-deployment   3/3     3            3           11s
candidate@node-1:~$ kubectl get pods -n cobra
NAME                              READY   STATUS    RESTARTS   AGE
www-deployment-d899c6b49-d6ccg    1/1     Running   0          14s
www-deployment-d899c6b49-f796l    1/1     Running   0          14s
www-deployment-d899c6b49-ztfcw    1/1     Running   0          14s
candidate@node-1:~$
```

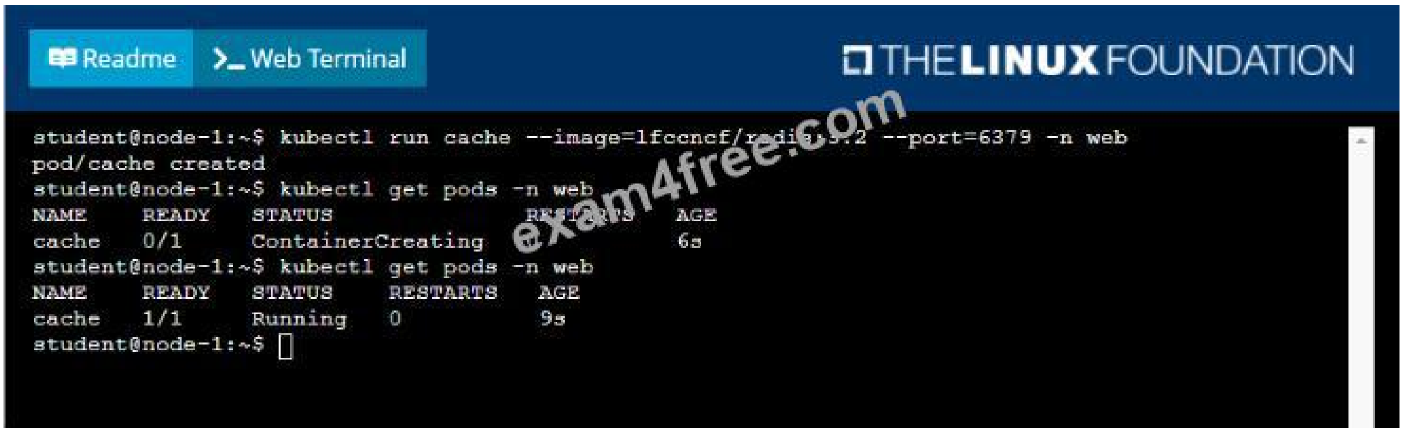
NEW QUESTION 26
Context



Set configuration context:

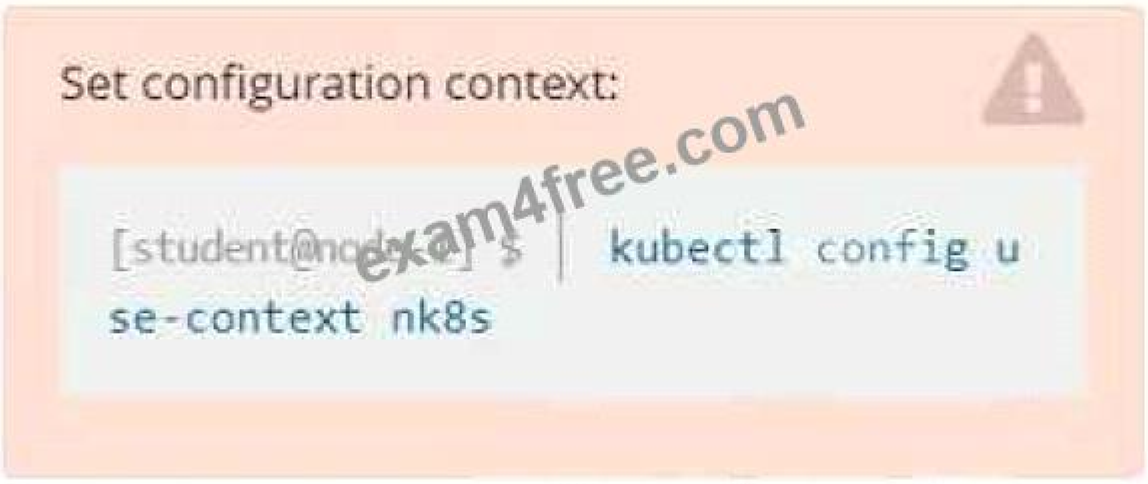
```
[student@node-1] $ | kubectl
config use-context k8s
```

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 Ifccncf/redis image with the 3.2 tag
* Expose port 6379
Answer:
Explanation:
Solution:



```
student@node-1:~$ kubectl run cache --image=lfccncf/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 27
Context



Set configuration context:

```
[student@node-1] $ | kubectl config u
se-context nk8s
```

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

Answer:

Explanation:
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 28
……