CKAD$^{Q\&As}$

Certified Kubernetes Application Developer (CKAD) Program

Pass Linux Foundation CKAD Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

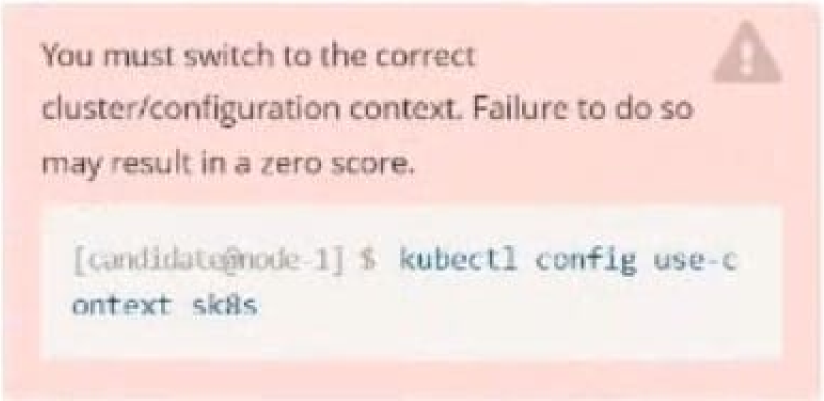
https://www.leads4pass.com/ckad.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Linux Foundation Official Exam Center

QUESTION 1

CORRECT TEXT



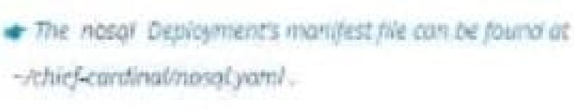
You must switch to the correct cluster/configuration context. Failure to do so may result in a zero score.

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

Task:

The pod for the Deployment named nosql in the crayfish namespace fails to start because its container runs out of resources.

Update the nosql Deployment so that the Pod:

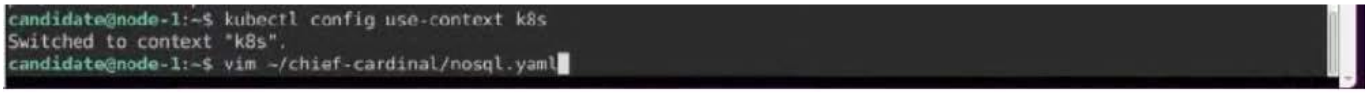


- The nosql Deployment's manifest file can be found at ~/chief-cardinal/nosql.yaml .

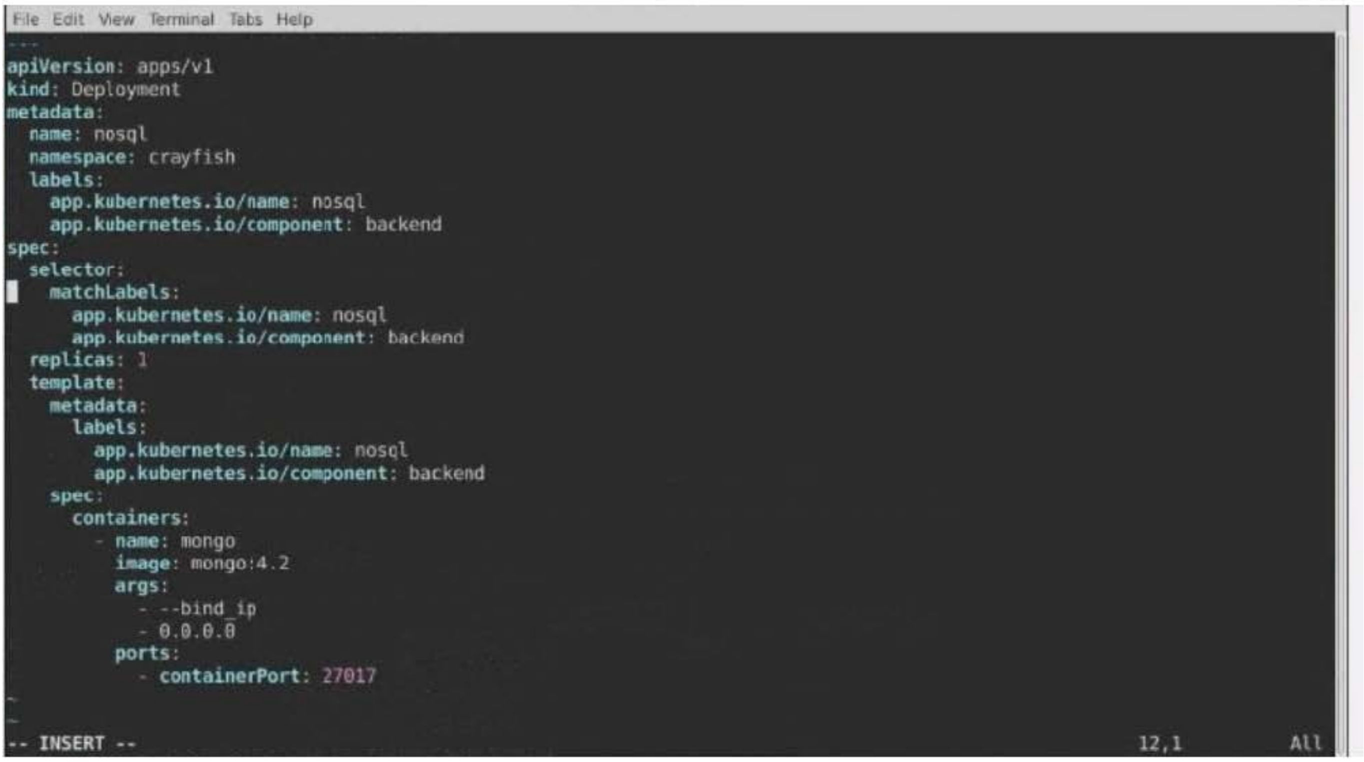
A. Please check explanations

B. Place Holder

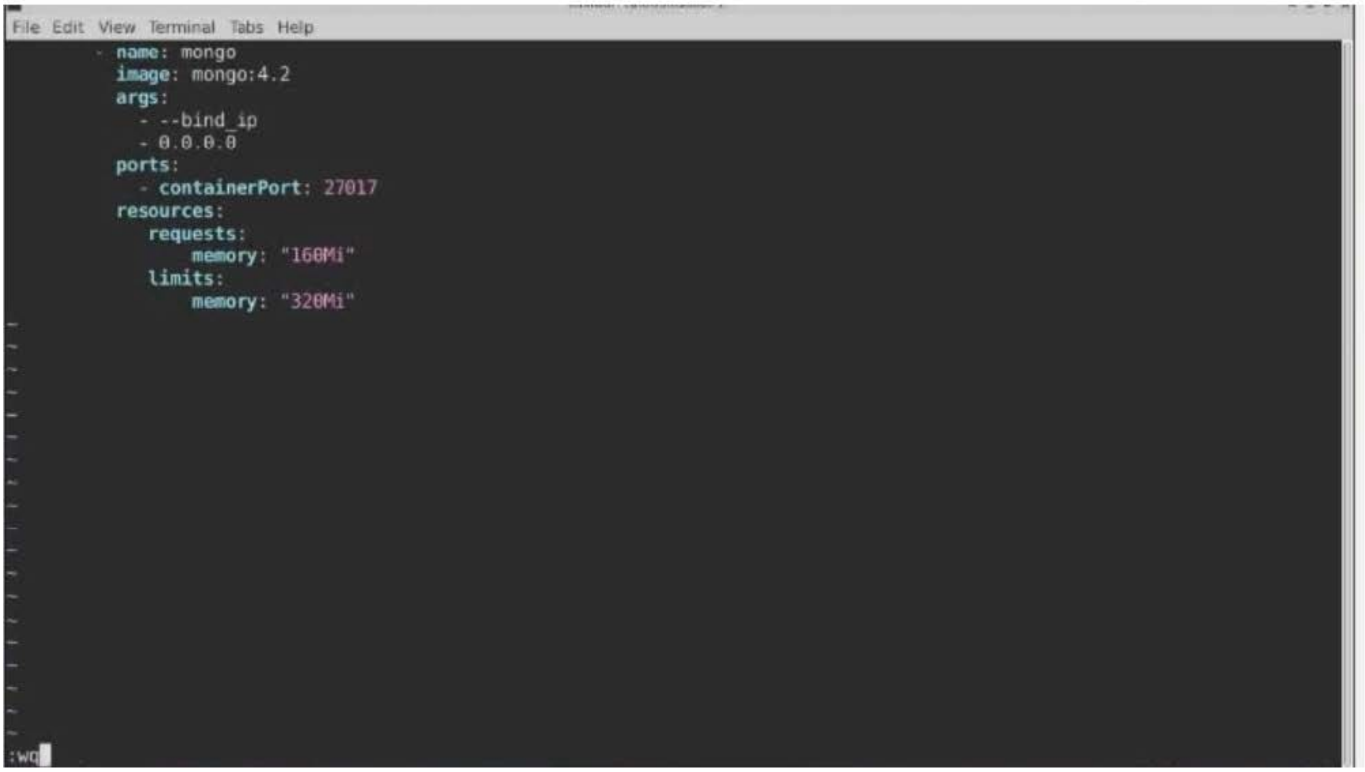
Correct Answer: A



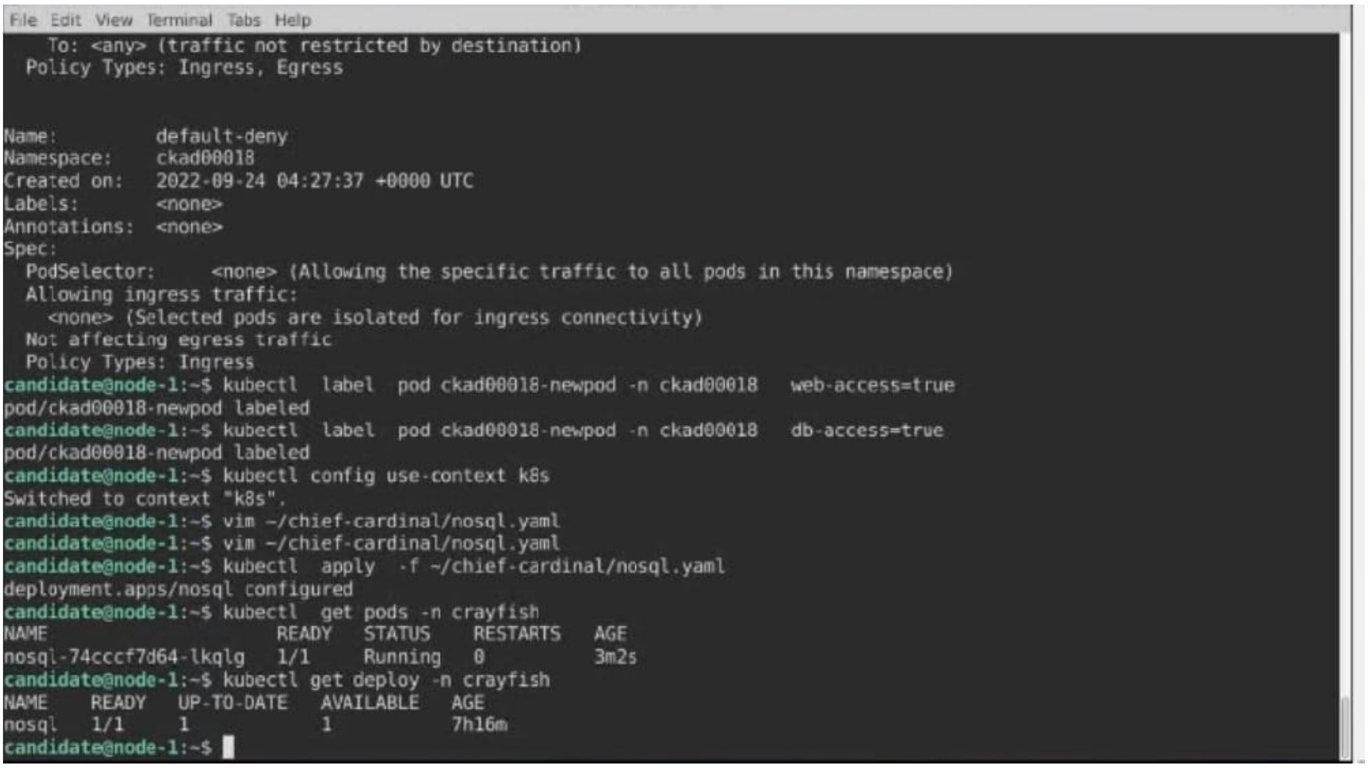
```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/chief-cardinal/nosql.yaml
```



```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nosql
  namespace: crayfish
  labels:
    app.kubernetes.io/name: nosql
    app.kubernetes.io/component: backend
spec:
  selector:
    matchLabels:
      app.kubernetes.io/name: nosql
      app.kubernetes.io/component: backend
  replicas: 1
  template:
    metadata:
      labels:
        app.kubernetes.io/name: nosql
        app.kubernetes.io/component: backend
    spec:
      containers:
        - name: mongo
          image: mongo:4.2
          args:
            - --bind_ip
            - 0.0.0.0
          ports:
            - containerPort: 27017
~
~
-- INSERT --                                                                    12,1          All
```



```
        - name: mongo
          image: mongo:4.2
          args:
            - --bind_ip
            - 0.0.0.0
          ports:
            - containerPort: 27017
          resources:
             requests:
                 memory: "160Mi"
             limits:
                 memory: "320Mi"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

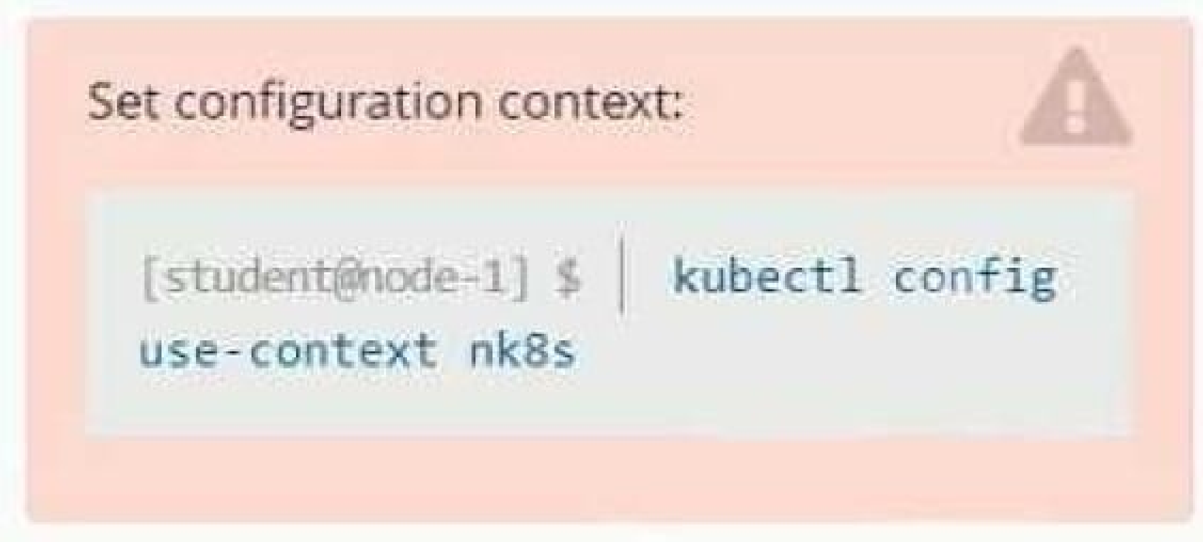


```
    To: <any> (traffic not restricted by destination)
  Policy Types: Ingress, Egress


Name:         default-deny
Namespace:    ckad00018
Created on:   2022-09-24 04:27:37 +0000 UTC
Labels:       <none>
Annotations:  <none>
Spec:
  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
candidate@node-1:~$ kubectl  label  pod ckad00018-newpod -n ckad00018   web-access=true
pod/ckad00018-newpod labeled
candidate@node-1:~$ kubectl  label  pod ckad00018-newpod -n ckad00018   db-access=true
pod/ckad00018-newpod labeled
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ vim ~/chief-cardinal/nosql.yaml
candidate@node-1:~$ vim ~/chief-cardinal/nosql.yaml
candidate@node-1:~$ kubectl  apply  -f ~/chief-cardinal/nosql.yaml
deployment.apps/nosql configured
candidate@node-1:~$ kubectl  get pods -n crayfish
NAME                     READY   STATUS    RESTARTS   AGE
nosql-74cccf7d64-lkqlg   1/1     Running   0          3m2s
candidate@node-1:~$ kubectl get deploy -n crayfish
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
nosql   1/1     1            1           7h16m
candidate@node-1:~$ 
```

QUESTION 2

CORRECT TEXT

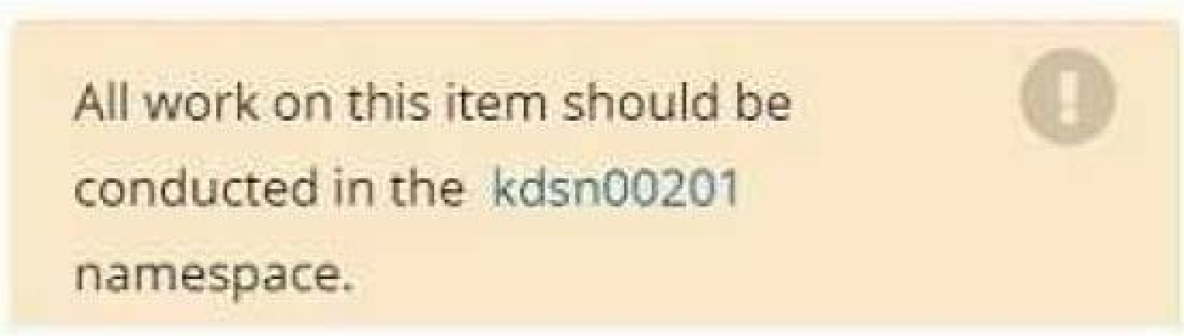


Set configuration context:

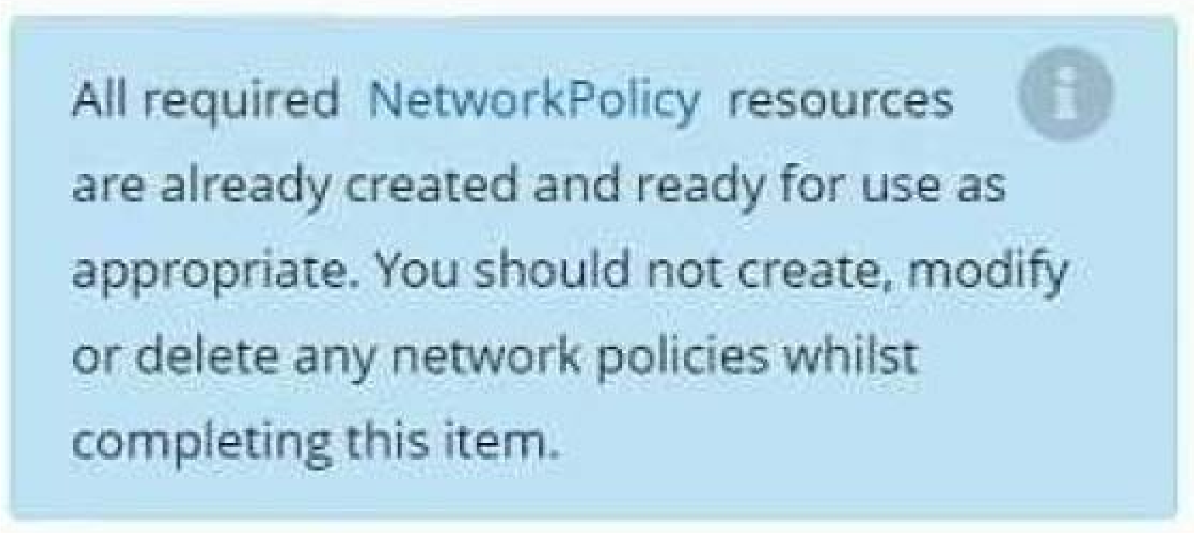
```
[student@node-1] $ | kubectl config use-context nk8s
```

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. Please check explanations

B. Place Holder

Correct Answer: A

```
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:

-

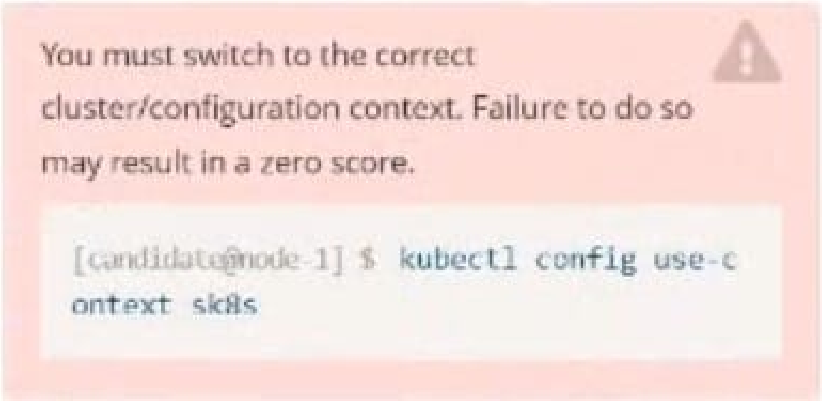
port: 53 protocol: UDP

-

port: 53 protocol: TCP

QUESTION 3

CORRECT TEXT



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

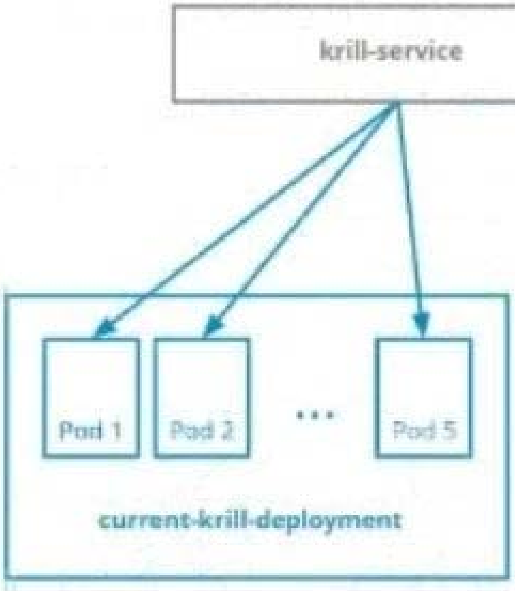
```
[candidate@node-1] $ kubectl config use-c
ontext sk8s
```

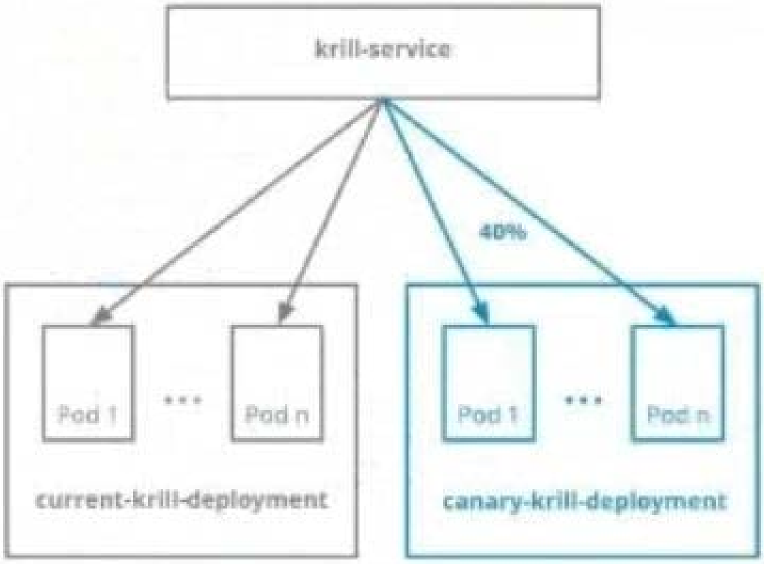
Context

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





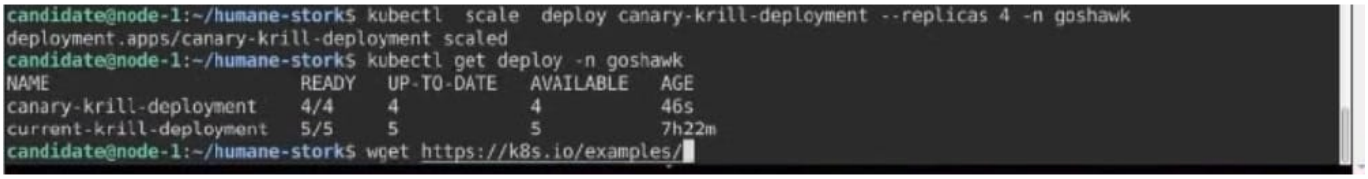
The Service is exposed on NodePort 30000 . To test its load-balancing, run:

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

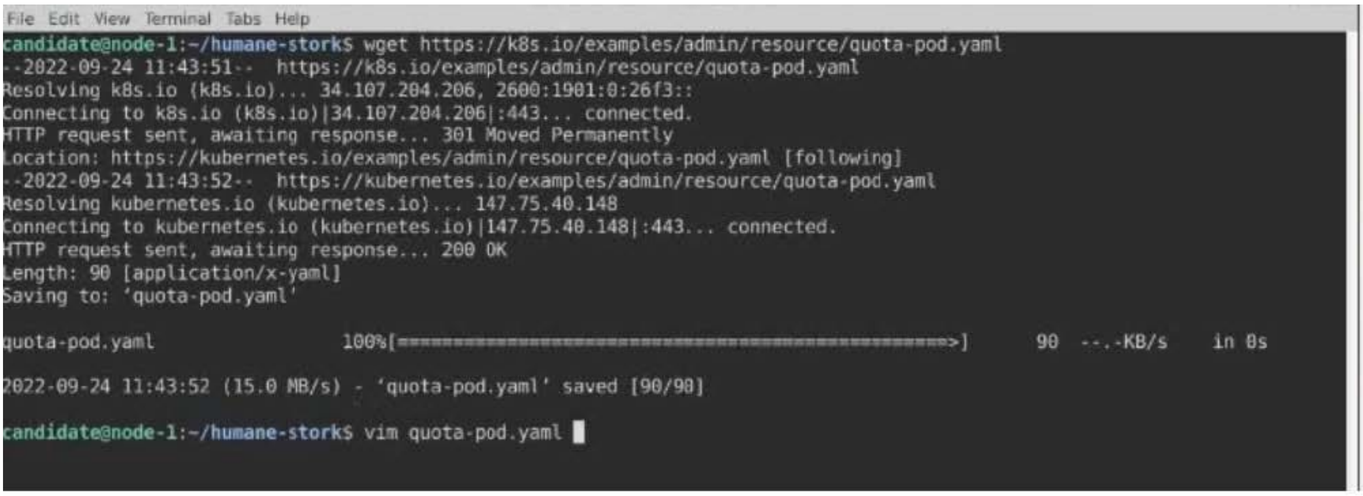
A. Please check explanations

B. Place Holder

Correct Answer: A



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

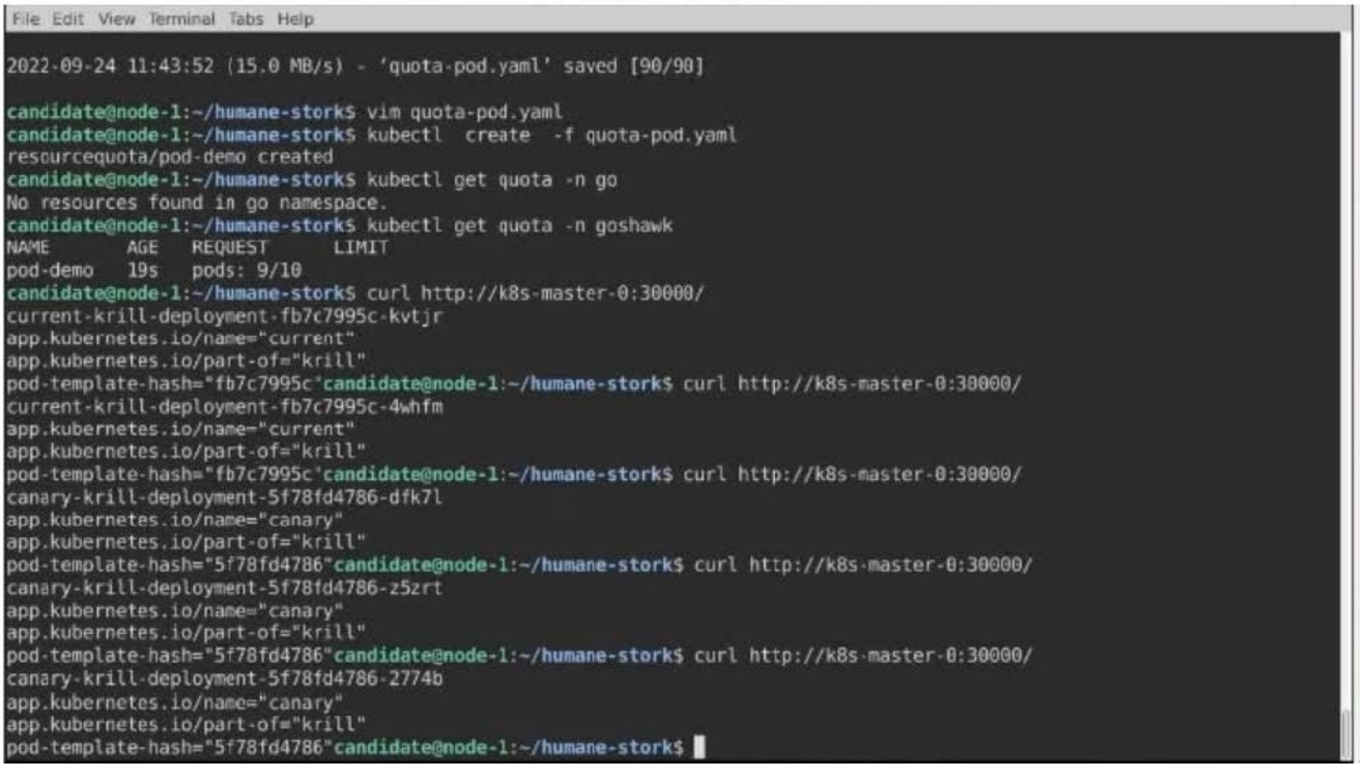


```
candidate@node-1:~/humane-stork$ wget https://k8s.io/examples/admin/resource/quota-pod.yaml
--2022-09-24 11:43:51--  https://k8s.io/examples/admin/resource/quota-pod.yaml
Resolving k8s.io (k8s.io)... 34.107.204.206, 2600:1901:0:26f3::
Connecting to k8s.io (k8s.io)|34.107.204.206|:443... connected.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

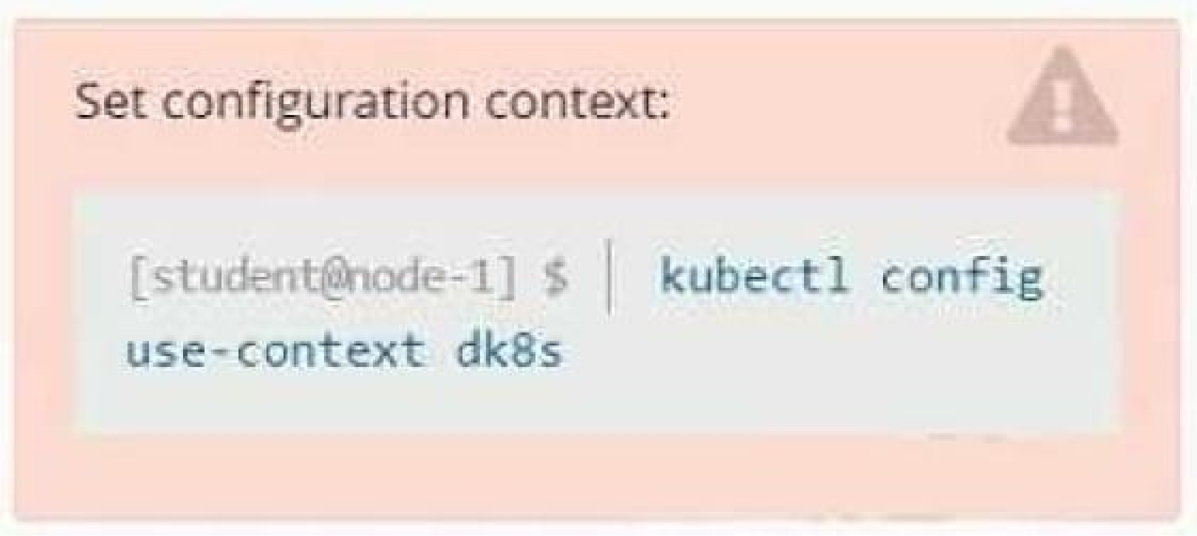


```
2022-09-24 11:43:52 (15.0 MB/s) - ‘quota-pod.yaml’ saved [90/90]

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-fb7c7995c-4whfm
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$
```

QUESTION 4

CORRECT TEXT



Set configuration context:

```
[student@node-1] $ | kubectl config
use-context dk8s
```

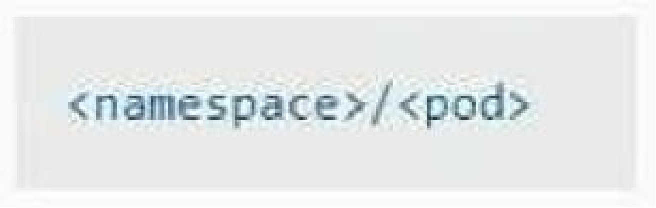
Context

A user has reported an application is unreachable due to a failing livenessProbe .

Task

Perform the following tasks:

Find the broken pod and store its name and namespace to /opt/KDOB00401/broken.txt in the format:



```
<namespace>/<pod>
```

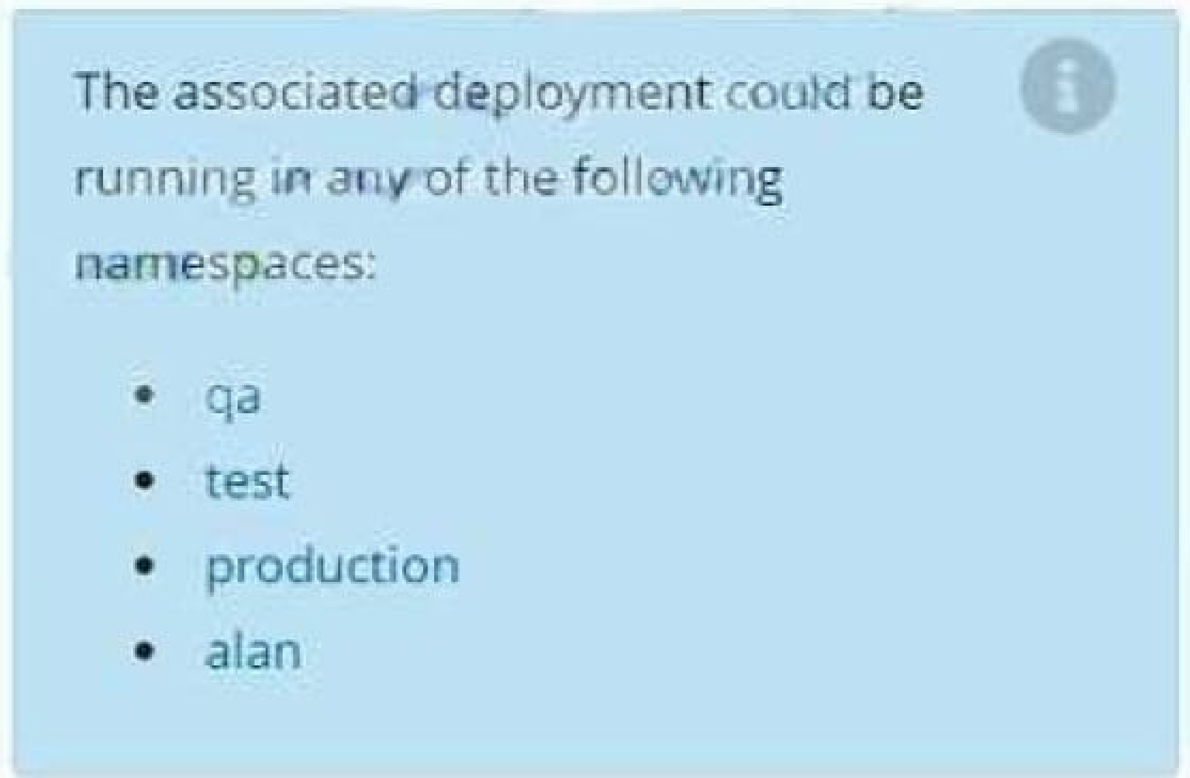
The output file has already been created

1.

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

2.

Fix the issue.



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

- qa
- test
- production
- alan

A. Please check explanations

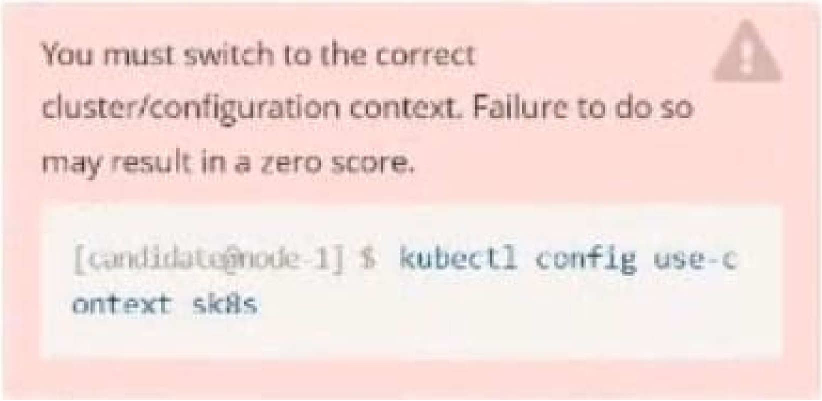
B. Place Holder

Correct Answer: A

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

QUESTION 5

CORRECT TEXT

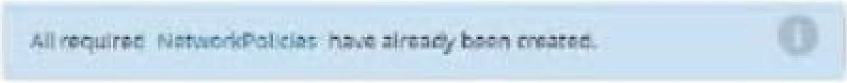


You must switch to the correct cluster/configuration context. Failure to do so may result in a zero score.

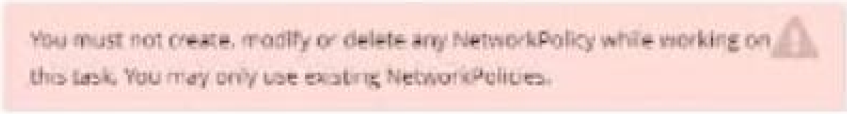
```
[candidate@node-1] $ kubectl config use-c
ontext sk8s
```

Task:

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



All required NetworkPolicies have already been created.

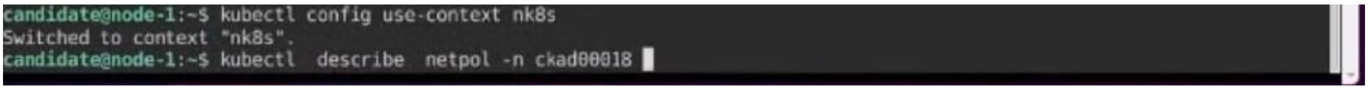


You must not create, modify or delete any NetworkPolicy while working on this task. You may only use existing NetworkPolicies.

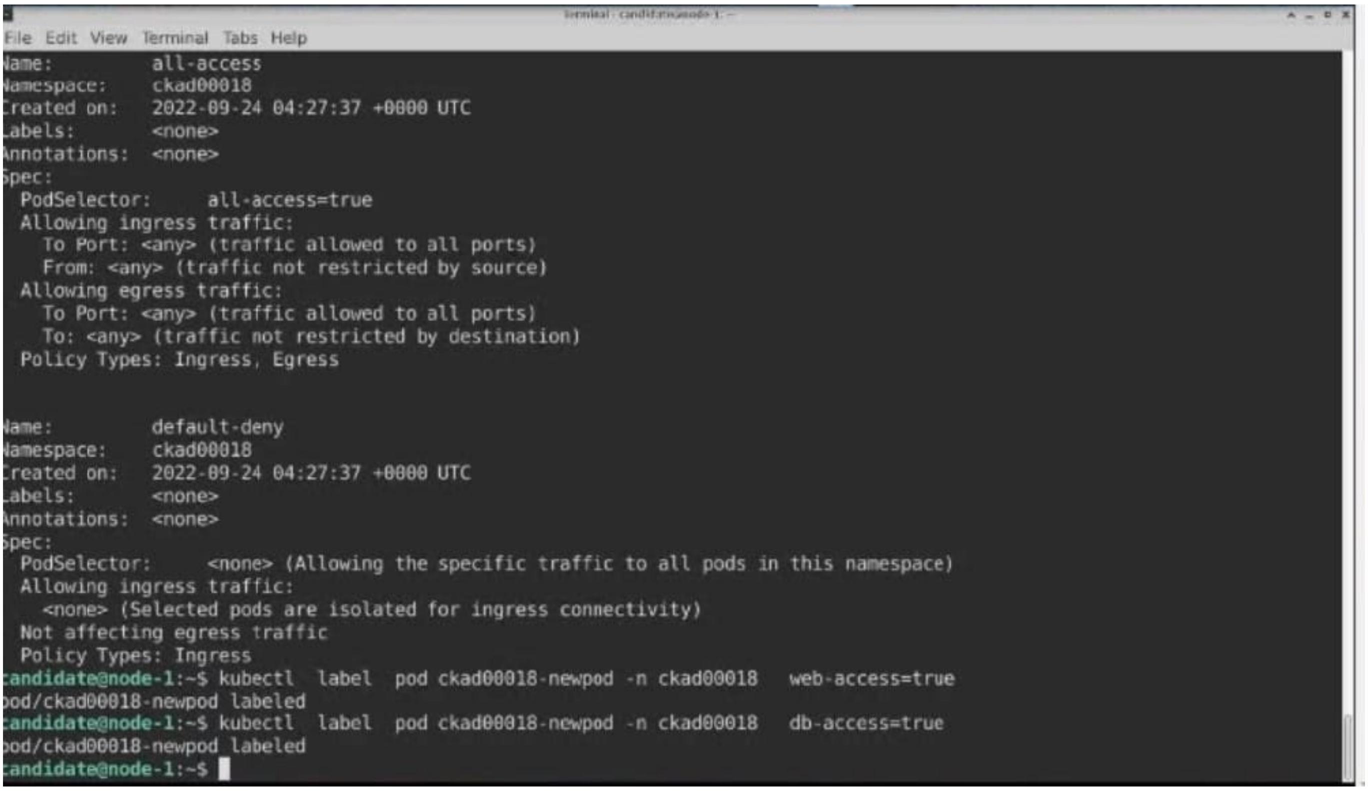
A. Please check explanations

B. Place Holder

Correct Answer: A



```
candidate@node-1:~$ kubectl config use-context nk8s
Switched to context "nk8s".
candidate@node-1:~$ kubectl  describe  netpol -n ckad00018
```



```
Name:         all-access
Namespace:    ckad00018
Created on:   2022-09-24 04:27:37 +0000 UTC
Labels:       <none>
Annotations:  <none>
Spec:
  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


Name:         default-deny
Namespace:    ckad00018
Created on:   2022-09-24 04:27:37 +0000 UTC
Labels:       <none>
Annotations:  <none>
Spec:
  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
candidate@node-1:~$ kubectl  label  pod ckad00018-newpod -n ckad00018   web-access=true
pod/ckad00018-newpod labeled
candidate@node-1:~$ kubectl  label  pod ckad00018-newpod -n ckad00018   db-access=true
pod/ckad00018-newpod labeled
candidate@node-1:~$
```

[CKAD VCE Dumps](#) [CKAD Practice Test](#) [CKAD Exam Questions](#)