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.geekcert.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

Context You are tasked to create a secret and consume the secret in a pod using environment variables as follow: Task

1.

Create a secret named another-secret with a key/value pair; key1/value4

2.

Start an nginx pod named nginx-secret using container image nginx, and add an environment variable exposing the value of the secret key key1, using COOL_VARIABLE as the name for the environment variable inside the pod

A. Please check explanations

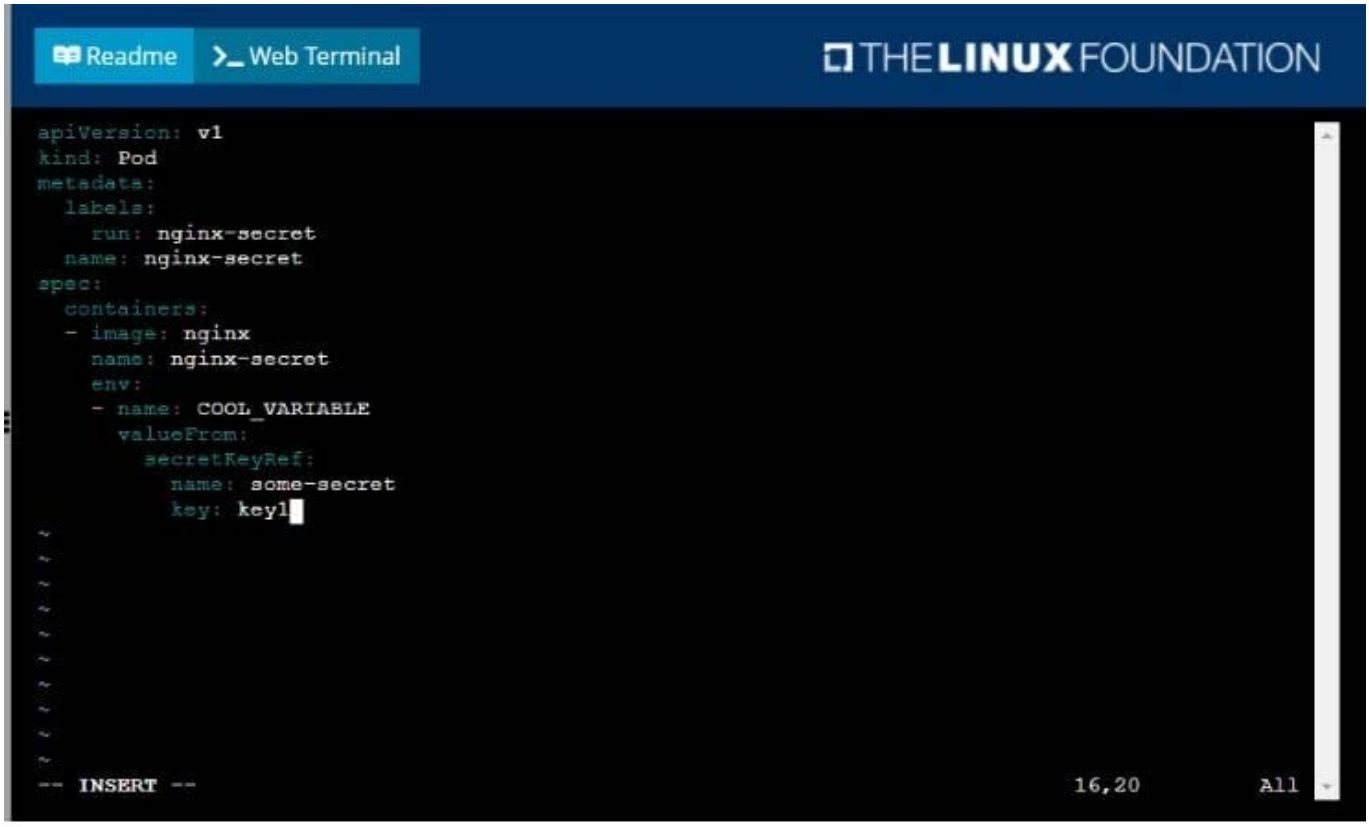
B. Place Holder

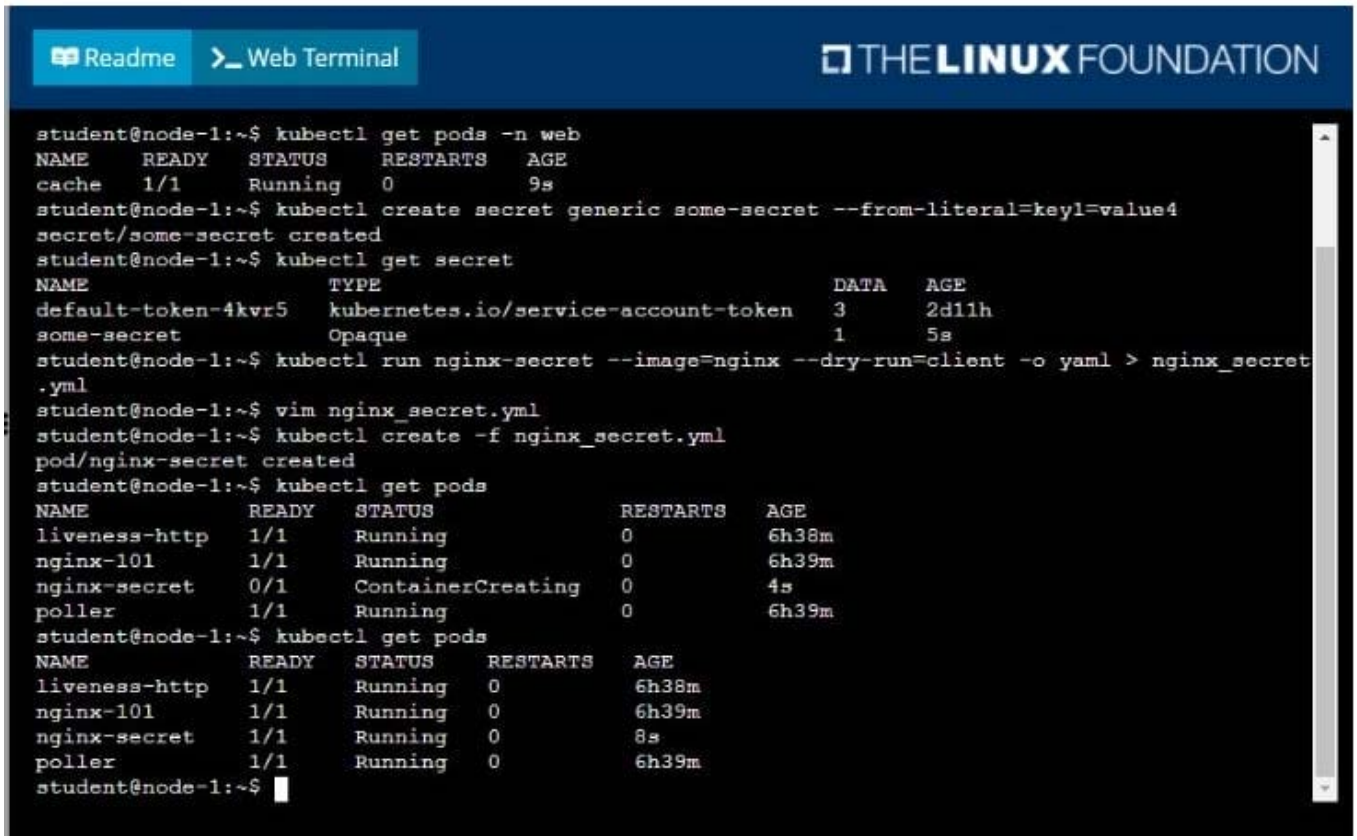
Correct Answer: A

```
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


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

QUESTION 2

CORRECT TEXT

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:

1.

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.

2.

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.

3.

Configure the probe-pod pod provided to use these endpoints

4.

The probes should use port 8080

A. Please check explanations

B. Place Holder

Correct Answer: A

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

QUESTION 3

CORRECT TEXT

Context

As a Kubernetes application developer you will often find yourself needing to update a running application.

Task

Please complete the following:

1.

Update the app deployment in the kdpd00202 namespace with a maxSurge of 5% and a maxUnavailable of 2%

2.

Perform a rolling update of the web1 deployment, changing the lfccncf/ngmx image version to 1.13

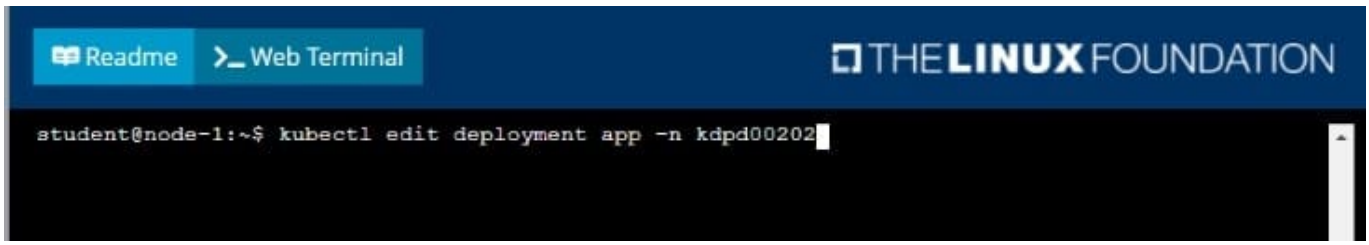
3.

Roll back the app deployment to the previous version

A. Please check explanations

B. Place Holder

Correct Answer: A



Readme | Web Terminal — THE LINUX FOUNDATION

```
  uid: 1dfa2527-5c61-46a9-8dd3-e24643d3ce14
spec:
  progressDeadlineSeconds: 600
  replicas: 10
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 5%
      maxUnavailable: 2
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
    spec:
      containers:
      - image: lfccncf/nginx:1.13
        imagePullPolicy: IfNotPresent
        name: nginx
        ports:
        - containerPort: 80
          protocol: TCP
:wq!
```

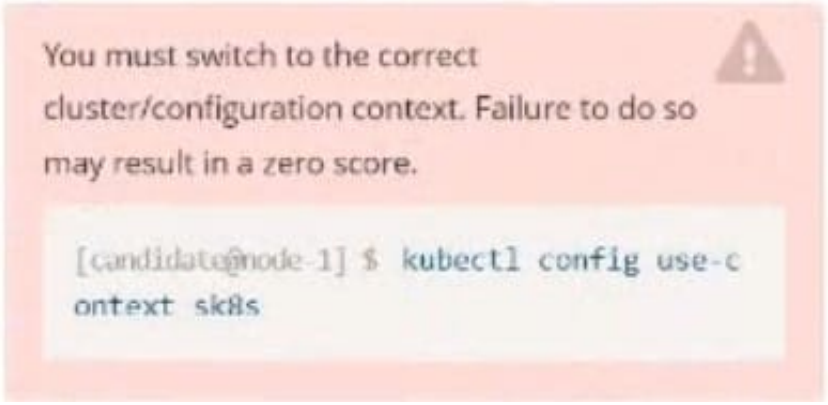
Readme | Web Terminal — THE LINUX FOUNDATION

```
student@node-1:~$ kubectl edit deployment app -n kdpd00202
deployment.apps/app edited
student@node-1:~$ kubectl rollout status deployment app -n kdpd00202
Waiting for deployment "app" rollout to finish: 6 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 7 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 7 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 7 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 8 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 8 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 8 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 8 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "app" rollout to finish: 8 of 10 updated replicas are available...
Waiting for deployment "app" rollout to finish: 9 of 10 updated replicas are available...
deployment "app" successfully rolled out
student@node-1:~$ kubectl rollout undo deployment app -n kdpd00202
deployment.apps/app rolled back
student@node-1:~$ kubectl rollout status deployment app -n kdpd00202
```

```
student@node-1:~$ kubectl rollout status deployment app -n kdpd00202
Waiting for deployment "app" rollout to finish: 6 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 6 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 6 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 6 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 7 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 7 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 9 out of 10 new replicas have been updated...
Waiting for deployment "app" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "app" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "app" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "app" rollout to finish: 8 of 10 updated replicas are available...
Waiting for deployment "app" rollout to finish: 9 of 10 updated replicas are available...
deployment "app" successfully rolled out
student@node-1:~$
```

QUESTION 4

CORRECT TEXT



Task:

Key3: value1

Add an environment variable named BEST_VARIABLE consuming the value of the secret key3.

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

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

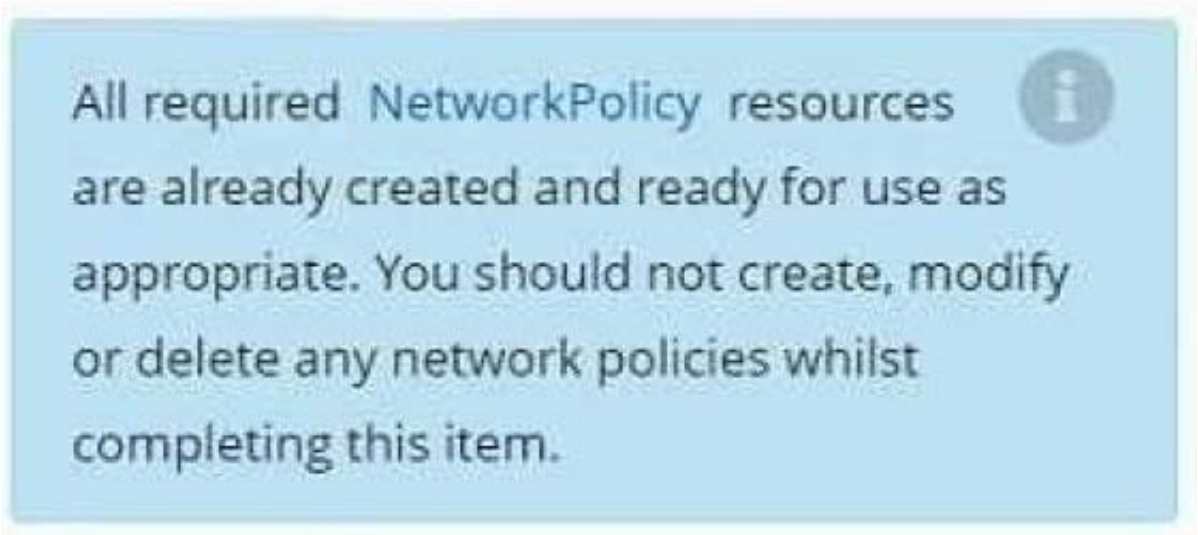
```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl  create  secret  generic  app-secret -n default  --from-literal=key3=value1
secret/app-secret created
candidate@node-1:~$ kubectl get secrets
NAME         TYPE     DATA   AGE
app-secret   Opaque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

QUESTION 5

CORRECT TEXT

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. 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