Linux-Foundation

Exam Questions CKAD

Certified Kubernetes Application Developer (CKAD) Program

About Exambible

Your Partner of IT Exam

Found in 1998

Exambible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, Exambible has its unique advantages that other companies could not achieve.

Our Advances

* 99.9% Uptime

All examinations will be up to date.

* 24/7 Quality Support

We will provide service round the clock.

* 100% Pass Rate

Our guarantee that you will pass the exam.

* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

NEW QUESTION 1
Exhibit:

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:
Key3: value1
Add an environment variable named BEST_VARIABLE consuming the value of the secret key3.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Solution:

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

Text Description automatically generated

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

Text Description automatically generated

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

NEW QUESTION 2
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. Mastered
B. Not Mastered

Answer: A

Explanation:
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
```

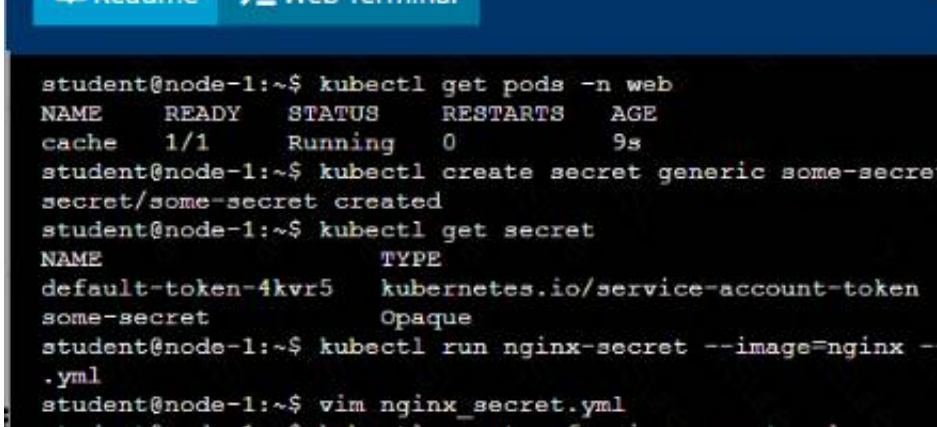
Readme | Web Terminal — THE LINUX FOUNDATION

```
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
```

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:~$
```

NEW QUESTION 3
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. Mastered
B. Not Mastered

Answer: A

Explanation:

Solution:

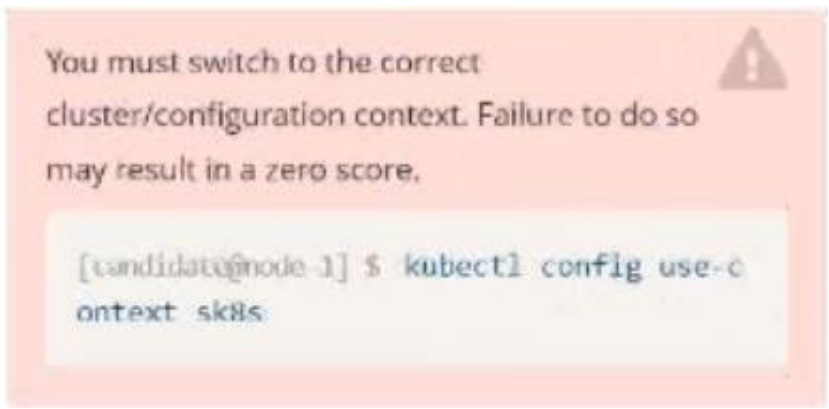
```
student@node-1:~$ kubectl create -f /opt/KDOB00201/counter.yaml
pod/counter created
student@node-1:~$ kubectl get pods
NAME              READY   STATUS    RESTARTS   AGE
counter           1/1     Running   0          10s
liveness-http     1/1     Running   0          6h45m
nginx-101         1/1     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:~$
```

```
student@node-1:~$ kubectl logs counter  > /opt/KDOB00201/log_output.txt
student@node-1:~$ kubectl logs counter  > /opt/KDOB00201/log_output.txt
student@node-1:~$ cat opt/KDOB00201/log_output.txt
```

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: 23fb22ee34d4272e4c9e005f1774515f
19: ec7e1a5d314da9a0ad45d53be5a7acae
20: 0bccdd8ee02cd42029e8162cd1c1197c
21: d6851ea43546216b95bcb81ced997102
22: 7ed9a38ea8bf0d86206569481442af44
23: 29b8416ddc63dbfcb987ab3c8198e9fe
24: 1f2062001df51a108ab25010f506716f
student@node-1:~$
```

NEW QUESTION 4
Exhibit:



Task:

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

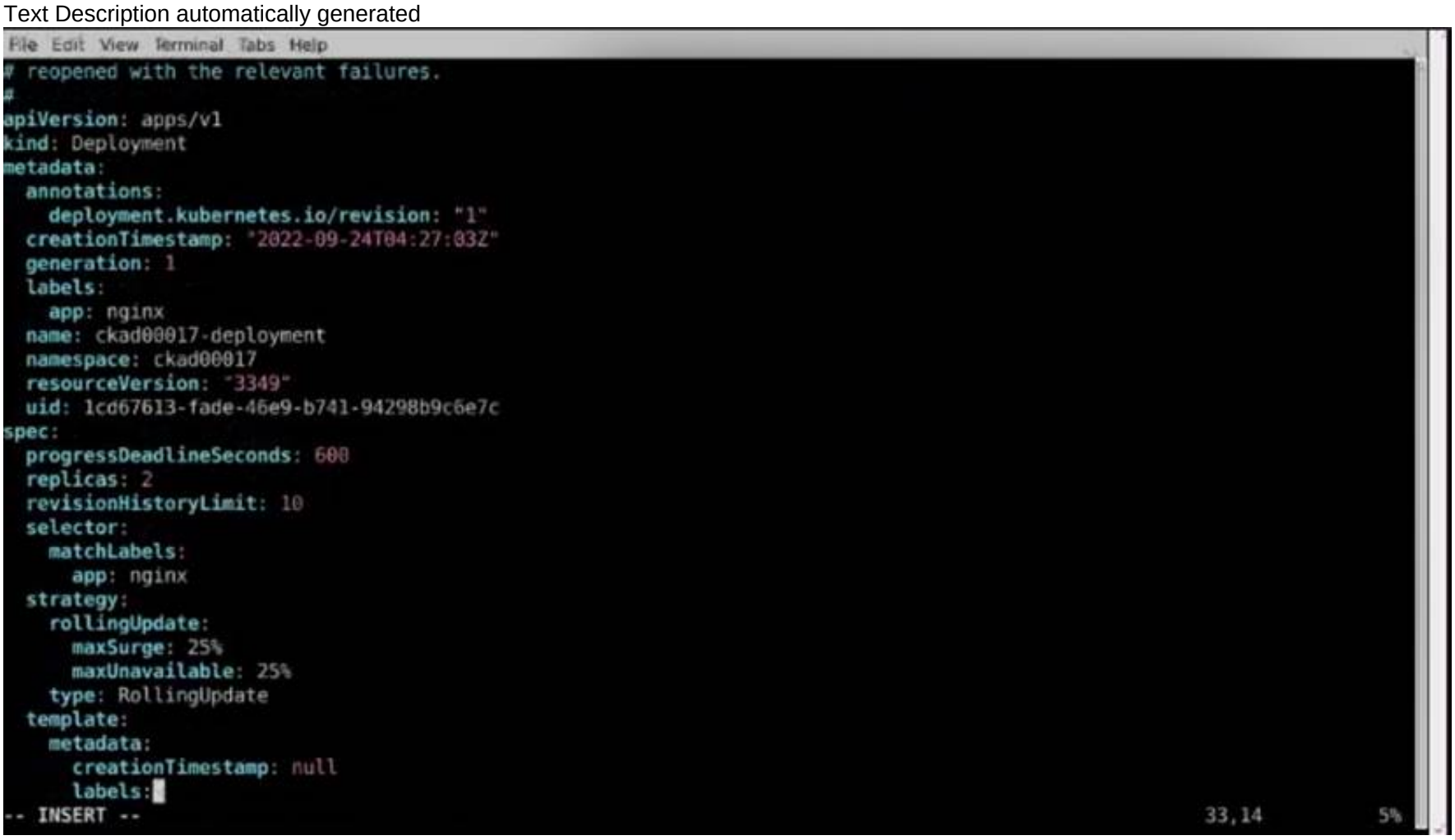
Role userUI

A. Mastered
B. Not Mastered

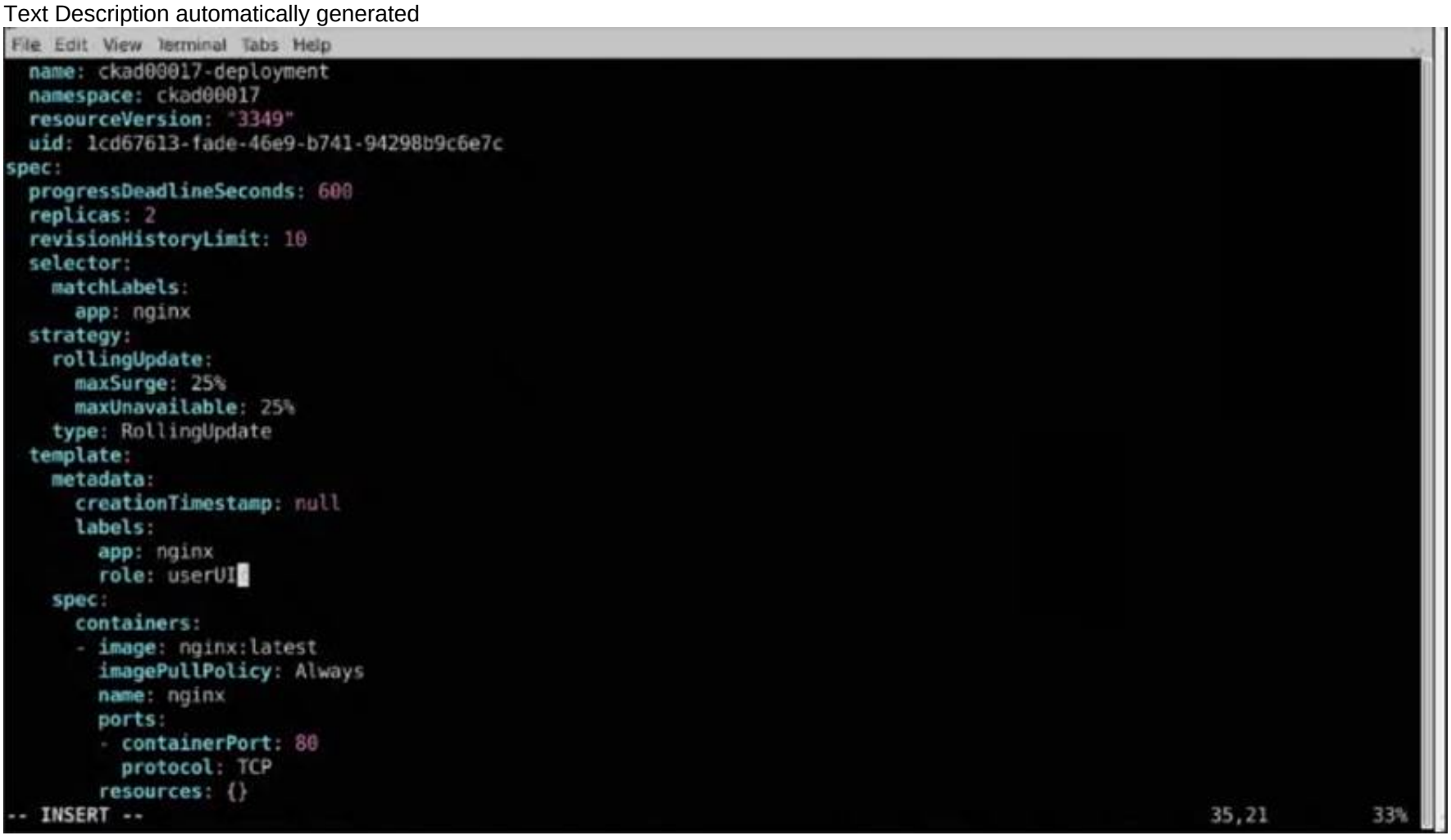
Answer: A

Explanation:

Solution:
Text Description automatically generated



Text Description automatically generated



Text Description automatically generated

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

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

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

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

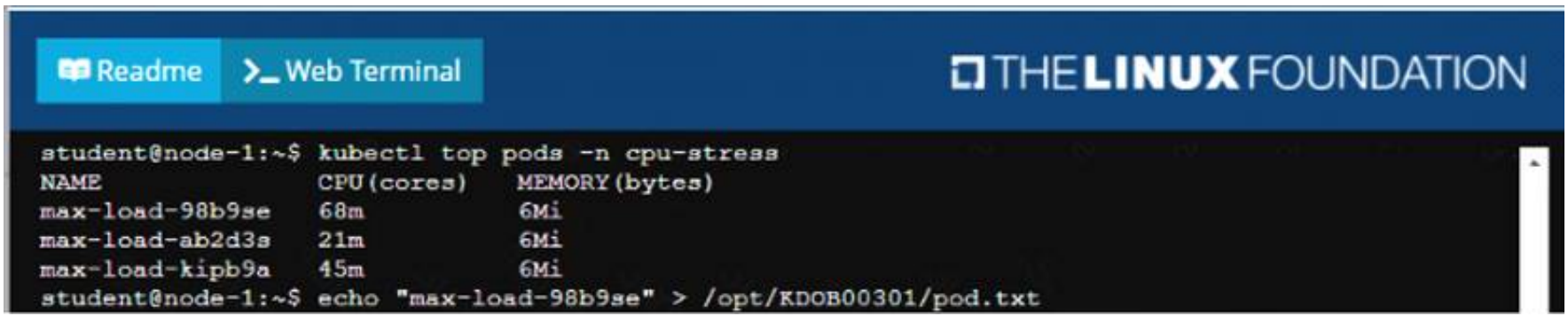
NEW QUESTION 5
Exhibit:

Context
It is always useful to look at the resources your applications are consuming in a cluster. Task
• From the pods running in namespace cpu-stress , write the name only of the pod that is consuming the most CPU to file /opt/KDOBG030l/pod.txt, which has already been created.

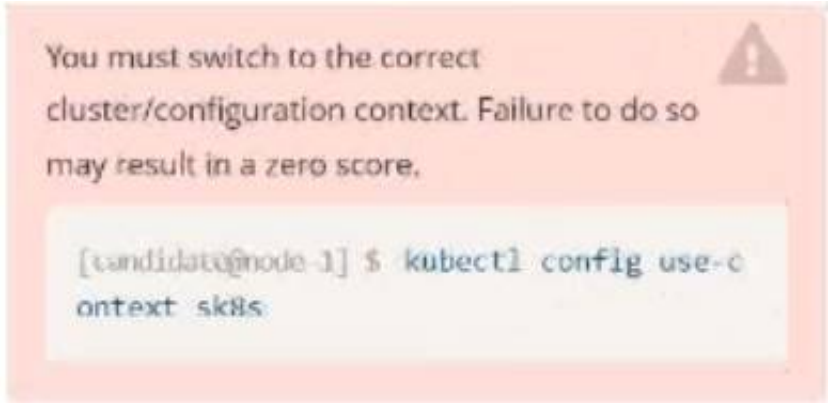
A. Mastered
B. Not Mastered

Answer: A

Explanation:
Solution:



NEW QUESTION 6
Exhibit:



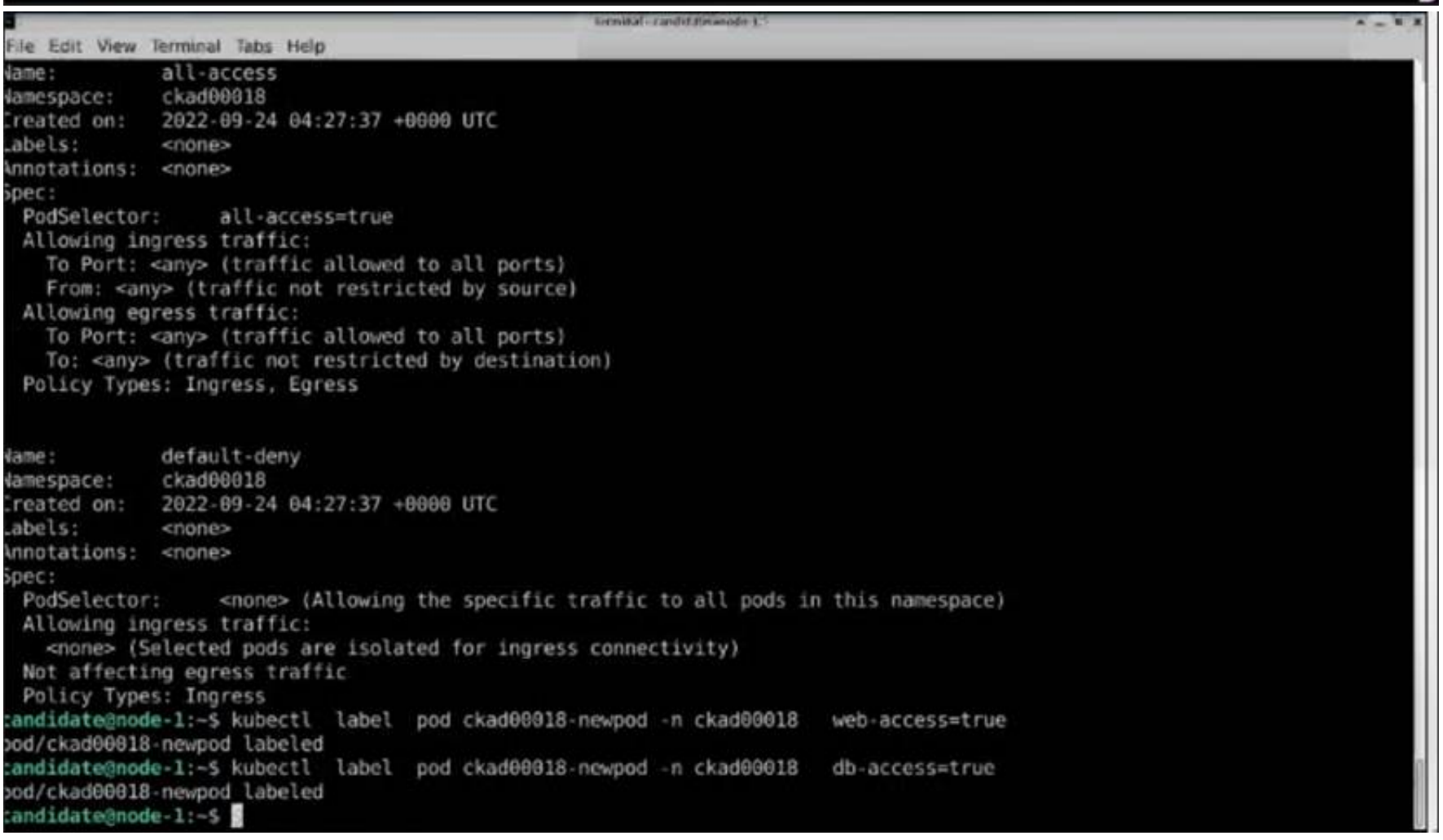
Task:
Update the Pod ckad00018-newpod in the ckad00018 namespace to use a NetworkPolicy allowing the Pod to send and receive traffic only to and from the pods web and db

A. Mastered
B. Not Mastered

Answer: A

Explanation:

Solution:



NEW QUESTION 7
Exhibit:

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

A. Mastered
B. Not Mastered

Answer: A

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 co
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 8
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. Mastered
B. Not Mastered

Answer: A

Explanation:
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
```

Readme | Web Terminal — THE LINUX FOUNDATION

```
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
```

Readme | Web Terminal — THE LINUX FOUNDATION

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

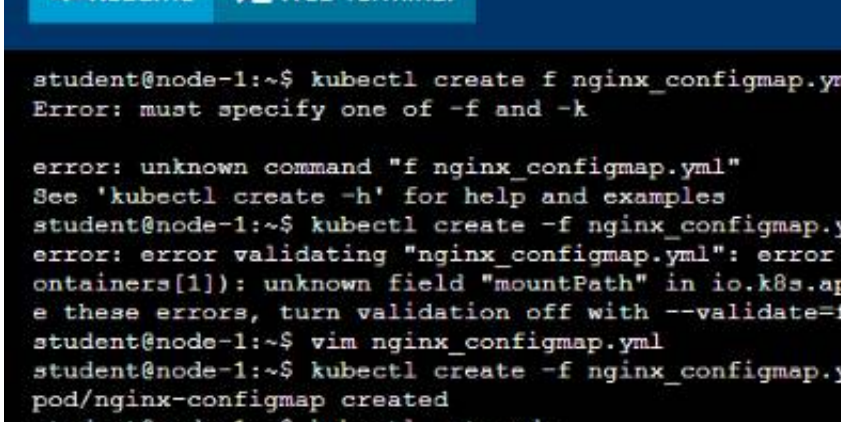
```
student@node-1:~$ kubectl create configmap another-config --from-literal=key4=value3
configmap/another-config created
student@node-1:~$ kubectl get configmap
NAME             DATA   AGE
another-config   1      5s
student@node-1:~$ kubectl run nginx-configmap --image=nginx --dry-run=client -o yaml > 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
```

Readme | Web Terminal — THE LINUX FOUNDATION

```
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
```

NEW QUESTION 9
Exhibit:



Task:
Create a Pod named nginx resources in the existing pod resources namespace. Specify a single container using nginx:stable image. Specify a resource request of 300m cpus and 1G1 of memory for the Pod's container.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Solution:

```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl  run nginx-resources -n pod-resources  --image=nginx:stable --dry-run=client -o yaml > hw.yaml
candidate@node-1:~$ vim hw.yaml
```

Text Description automatically generated with medium confidence

```
File  Edit  View  Terminal  Tabs  Help
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx-resources
  name: nginx-resources
  namespace: pod-resources
spec:
  containers:
  - image: nginx:stable
    name: nginx-resources
    resources:
      requests:
          cpu: 300m
          memory: "1Gi"
~
~
~
~
~
~
~
~
~
~
~
:wq
```

Text Description automatically generated

```
candidate@node-1:~$ kubectl config use-context k8s
Switched to context "k8s".
candidate@node-1:~$ kubectl  run nginx-resources -n pod-resources  --image=nginx:stable --dry-run=client -o yaml > hw.yaml
candidate@node-1:~$ vim hw.yaml
candidate@node-1:~$ kubectl  create  -f hw.yaml
pod/nginx-resources created
candidate@node-1:~$ kubectl get pods -n pod-resources
NAME              READY   STATUS    RESTARTS   AGE
nginx-resources   1/1     Running   0          13s
candidate@node-1:~$ kubectl  describe  pods -n pod-resources
```

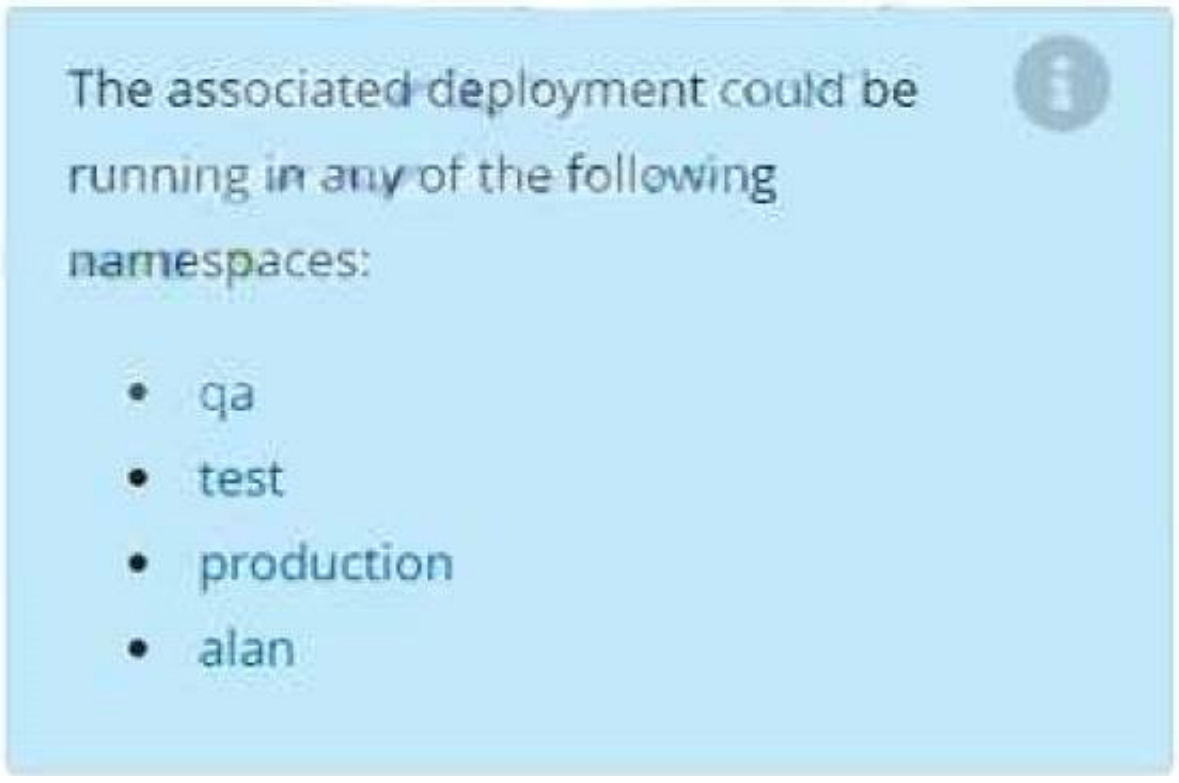
Text Description automatically generated

```
File  Edit  View  Terminal  Tabs  Help
      memory:     1Gi
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-dmx9j (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  kube-api-access-dmx9j:
    Type:                     Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:   3607
    ConfigMapName:            kube-root-ca.crt
    ConfigMapOptional:        <nil>
    DownwardAPI:              true
QoS Class:                    Burstable
Node-Selectors:               <none>
Tolerations:                  node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                              node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  20s   default-scheduler  Successfully assigned pod-resources/nginx-resources to k8s-node-0
  Normal  Pulling    19s   kubelet            Pulling image "nginx:stable"
  Normal  Pulled     13s   kubelet            Successfully pulled image "nginx:stable" in 6.55664052s
  Normal  Created    13s   kubelet            Created container nginx-resources
  Normal  Started    12s   kubelet            Started container nginx-resources
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>
```

NEW QUESTION 10
Exhibit:

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:

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.



A. Mastered
B. Not Mastered

Answer: A

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 10
Exhibit:

Task:
The pod for the Deployment named nosql in the craytisn namespace fails to start because its container runs out of resources.
Update the nosol Deployment so that the Pod:

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

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Solution:

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

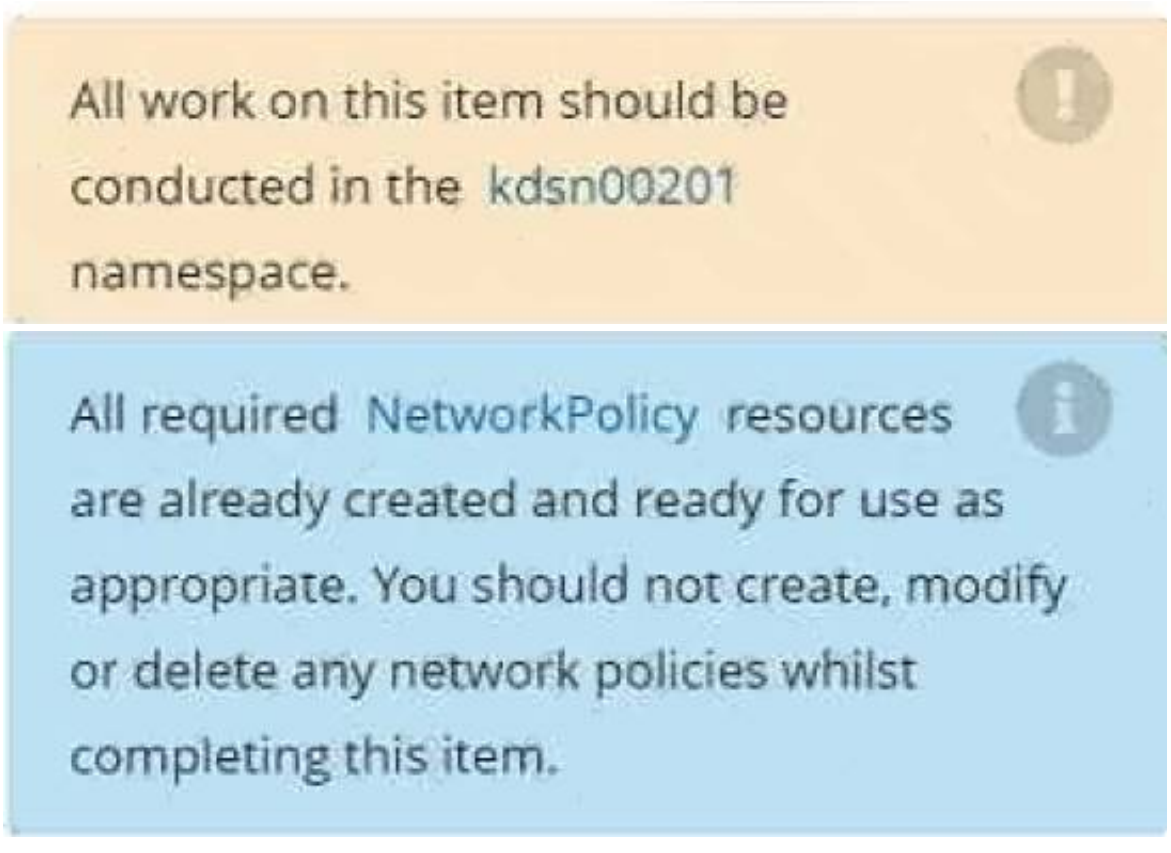
```
File  Edit  View  Terminal  Tabs  Help
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nosql
  namespace: crayfish
  labels:
    app.kubernetes.io/name: nosql
    app.kubernetes.io/component: backend
spec:
  selector:
    matchLabels:
      app.kubernetes.io/name: nosql
      app.kubernetes.io/component: 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
```

```
File  Edit  View  Terminal  Tabs  Help
      - name: mongo
        image: mongo:4.2
        args:
          - --bind_ip
          - 0.0.0.0
        ports:
          - containerPort: 27017
        resources:
          requests:
              memory: "160Mi"
          limits:
              memory: "320Mi"
:wq
```

```
File  Edit  View  Terminal  Tabs  Help
    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:~$
```

NEW QUESTION 15
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. Mastered
B. Not Mastered

Answer: A

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 19
.....

Relate Links

100% Pass Your CKAD Exam with Exambible Prep Materials

https://www.exambible.com/CKAD-exam/

Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - https://www.exambible.com/