

Free Questions for CKAD by dumpssheet

Shared by Flynn on 22-07-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Refer to 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.

Options:

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

```
File Edit View Terminal Tabs Help
apiVersion: v1
kind: Pod
netadata:
 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"
```

:wa

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 θ 13s candidate@node-1:~\$ kubectl describe pods -n pod-resources

File Edit V	liew Termina	Tabs H	elp		
mem	ory: 1	Gi			
Envir	onment: <	none>			
Mount	s:				
/va	r/run/secr	ets/kub	ernetes.io/servicea	ccount from kube-api-access-dmx9j (ro)	
Condition	s:	10000			
Туре		Status			
Initial	ized	True			
Ready		True			
Contain	ersReady	True			
PodSche	duled	True			
Volumes:		101201			
kube-ap	1-access-0	1mx9]:	- Alexandra - Alexandra		
Type:			Projected (a vol	ume that contains injected data from multiple sources)	
Token	Expiration	Seconds	: 3607		
Conti	gMapName:		kube-root-ca.crt		
Conti	gMapuption	ial:	<n11></n11>		
Downw	ardAP1:		true		
Qos class			Burstable		
Node-Sele	ctors:		<none></none>	the factor of the Principle and Principle Street Stre	
Toteratio	ns:		node.kubernetes.io/not-ready:NoExecute op=Exists for 300s		
node.kubernetes.io/unreachable:NoExecute		node.kubernetes.	10/Unreachable:NoExecute op=Exists for 3005		
Events:	Deseas	244	E and		
туре	Reason	Age	From	nessage	
Normal	Cabadulas	204	default schedules	Euconstitution and and recourses (aniou recourses to b0s and 0	
Normal	Bulling	100	kubalat	Dulling image "ngiovistable"	
Normal	Pulled	126	kubelet	Successfully pulled image "priny stable" in 6 55664052s	
Normal	Created	120	kubalat	Croated container printy recourses	
Normal	Started	126	kubalat	Started container nginx resources	
candidate	Starteu	LLS	L config uso contos	+ Loe	
Switched	to context	"kRe"	c coning use-contex		
candidate	anode-1:-	kubect	create denlow e	xpose -n_ckad00014image lfccncf/noinx:1 13 7dry-run=client -n_yamla	
contractor	Sugar Ti -	HUDGEL	elegre deproy e	Apose in chadooda aninge creenerriganization of y fun-celene o yaines	

А

Question 2

Question Type: MultipleChoice

Refer to Exhibit.



Task:

1) First update the Deployment cka00017-deployment in the ckad00017 namespace:

*To run 2 replicas of the pod

*Add the following label on the pod:

Role userUI

2) Next, Create a NodePort Service named cherry in the ckad00017 nmespace exposing the ckad00017-deployment Deployment on TCP port 8888

Options:

A- Explanation:

Solution:

```
File Edit View Terminal Tabs Help
# reopened with the relevant failures.
apiVersion: apps/v1
kind: Deployment
metadata:
 annotations:
    deployment.kubernetes.io/revision: "1"
 creationTimestamp: "2022-09-24T04:27:03Z"
 generation: 1
  labels:
   app: nginx
 name: ckad00017-deployment
 namespace: ckad00017
 resourceVersion: "3349"
 uid: 1cd67613-fade-46e9-b741-94298b9c6e7c
spec:
 progressDeadlineSeconds: 600
 replicas: 2
 revisionHistoryLimit: 10
 selector:
   matchLabels:
      app: nginx
 strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
 template:
   metadata:
      creationTimestamp: null
      labels:
 - INSERT --
                                                                                                         33,14
                                                                                                                         5%
```

```
File Edit View Terminal Tabs Help
 name: ckad00017-deployment
 namespace: ckad00017
 resourceVersion: "3349"
 uid: 1cd67613-fade-46e9-b741-94298b9c6e7c
spec:
 progressDeadlineSeconds: 600
 replicas: 2
 revisionHistoryLimit: 10
 selector:
   matchLabels:
     app: nginx
 strategy:
   rollingUpdate:
     maxSurge: 25%
     maxUnavailable: 25%
   type: RollingUpdate
 template:
   metadata:
     creationTimestamp: null
     labels:
       app: nginx
       role: userUI
   spec:
     containers:
     - image: nginx:latest
       imagePullPolicy: Always
       name: nginx
       ports:
       - containerPort: 80
         protocol: TCP
       resources: {}
-- INSERT --
```

File Edit View Terminal Tabs Help backend-deployment-59d449b99d-h2zjg Running 0/1 0 9s backend-deployment-78976f74f5-b8c85 1/1 Θ 6h40m Running backend-deployment-78976f74f5-flfsi 1/1 Running Θ 6h40m candidate@node-1:~\$ kubectl get deploy -n staging NAME READY UP-TO-DATE AVAILABLE AGE backend-deployment 3/3 6h40m candidate@node-1:~\$ kubectl get deploy -n staging NAME UP-TO-DATE AVAILABLE AGE READY 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 AGE RESTARTS buffalo-deployment-776844df7f-r5fsb 1/1 Running 6h38m 0 buffalo-deployment-859898c6f5-zx5gj 0/1 ContainerCreating Θ 8s candidate@node-1:~\$ kubectl get deploy -n gorilla NAME UP-TO-DATE AVAILABLE READY AGE buffalo-deployment 1/1 6h38m 1 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:~\$

File Edit View Terminal Tabs Help		
candidate@node-1:-\$ kubectl get pods -n gorilla		65 s.
NAME READY STATUS	RESTARTS A	GE
buffalo-deployment-776844df7f-r5fsb 1/1 Running	Θ 6	h38m
<pre>buffalo-deployment-859898c6f5-zx5gj 0/1 ContainerCreati</pre>	ng Θ 8	S
<pre>candidate@node-1:~\$ kubectl get deploy -n gorilla</pre>		
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".	0.000000000	
candidate@node-1:~\$ kubectl edit deploy ckad00017-deployment	-n ckad00017	
deployment.apps/ckad0001/-deployment edited		
candidate@node-1:~\$ Kubecti expose deploy cKaduuu1/-deploym	ent -n ckaduuul	
candidate@node_lt_5_kubactl_avaaca_danlov_ckad00017_danlovm	not a chadaaa	
ckad00014 ckad00015 ckad00017	Enc -n cka00001	
candidateGnode-1:~\$ kubectl expose deploy ckad00017-deploym	ent -n ckad0001	
ckad00014 ckad00015 ckad00017	cire on courses	
candidate@node-1:-\$ kubectl expose deploy ckad00017-deploym	ent -n ckad0001	
ckad00014 ckad00015 ckad00017		
candidate@node-1:~\$ kubectl expose deploy ckad00017-deploym	ent -n ckad0001	
ckad00014 ckad00015 ckad00017		
candidate@node-1:-\$ kubectl expose deploy ckad00017-deploym	ent -n ckad0001	
ckad00014 ckad00015 ckad00017		
<pre>candidate@node-1:-\$ kubectl expose deploy ckad00017-deploym</pre>	ent -n ckad0001	
ckad00014 ckad00015 ckad00017		
<pre>candidate@node-1:~\$ kubectl expose deploy ckad00017-deploym</pre>	ent -n ckad0001	
ckad00014 ckad00015 ckad00017		
candidate@node-1:~\$ kubectl expose deploy ckad00017-deploym	ent -n ckad0001	
ckadeeela ckadeeela daalaa ahalaa ahalaa ahalaa ahalaa	Troophalla a data	The second s
candidategnode-1:~S Rubecti expose deploy ckad0001/-deploym	ent -n ckad0001/	name=cherryport=8888type=ModePort
candidate@node_1:_s		그는 그는 것 같아요. 그는 것 같아요. 그는 것
calluida cegilode - 1>		

pe=N

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 CLUSTER-IP EXTERNAL-IP PORT(S) TYPE 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:~\$

А

Question 3

Question Type: MultipleChoice

Refer to Exhibit.



Task:

A pod within the Deployment named buffale-deployment and in namespace gorilla is logging errors.

1) Look at the logs identify errors messages.

Find errors, including User "system:serviceaccount:gorilla:default" cannot list resource "deployment" [...] in the namespace "gorilla"

2) Update the Deployment buffalo-deployment to resolve the errors in the logs of the Pod.

The buffalo-deployment 'S manifest can be found at -/prompt/escargot/buffalo-deployment.yaml

Options:

A- Explanation:

Solution:

File Edit View Terminal Tabs Help deployment.apps/backend-deployment configured candidate@node-1:-\$ kubectl get pods -n staging NAME READY STATUS RESTARTS AGE backend-deployment-59d449b99d-cxct6 1/1 Running Θ 20s backend-deployment-59d449b99d-h2zig 95 $\theta/1$ Running Θ backend-deployment-78976f74f5-b8c85 1/1 Θ 6h40m Running 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 6h40m candidate@node-1:~\$ kubectl get deploy -n staging NAME UP-TO-DATE AVAILABLE AGE READY backend-deployment 3/3 6h41m 3 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 AGE STATUS RESTARTS buffalo-deployment-776844df7f-r5fsb 1/1 Running 6h38m Θ buffalo-deployment-859898c6f5-zx5gj ContainerCreating 0/1 8s 0 candidate@node-1:~\$ kubectl get deploy -n gorilla NAME UP-TO-DATE AVAILABLE READY AGE buffalo-deployment 1/1 1 6h38m candidate@node-1:~\$

```
File Edit View Terminal Tabs Help
```

```
candidate@node-1:~$ vi ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:-$ kubectl config use-context sk8s
Switched to context "sk8s".
candidate@node-1:~$ vim .vimrc
candidate@node-1:-$ vim -/spicy-pikachu/backend-deployment.yaml
candidate@node-1:-$ kubectl apply .f -/spicy-pikachu/backend-deployment.yaml
deployment.apps/backend-deployment configured
candidate@node-1:~$ kubectl get pods -n staging
NAME
                                    READY STATUS RESTARTS AGE
backend-deployment-59d449b99d-cxct6 1/1
                                            Running 0
backend-deployment-59d449b99d-h2zjq 0/1
                                            Running
backend-deployment-78976f74f5-b8c85 1/1
                                            Running 0
                                                                6h40m
                                            Running 0
backend-deployment-78976f74f5-flfsj 1/1
                                                                6h40m
candidate@node-1:~$ kubectl get deploy -n staging
NAME
                   READY UP-TO-DATE AVAILABLE
                                                    AGE
backend-deployment 3/3
                                                    6h40m
candidate@node-1:-$ kubectl get deploy -n staging
NAME
                   READY UP-TO-DATE AVAILABLE AGE
backend-deployment 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 go
```

File Edit View Terminal Tabs Help	14.16.14.14.14.16.14.16.16.16.16.16.16.16.16.16.16.16.16.16.				
deployment apps/backend.deployment configured					
candidate@node_1:-\$ kubect1 get node _n staging					
NAME READY STATUS	RESTARTS AGE				
backend-deployment-59d449b99d-cxct6 1/1 Running	0 205				
backend-deployment-59d449b99d-h2zig 0/1 Running	0 95				
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 AG	βE				
backend-deployment 3/3 3 6h	140m				
candidate@node-1:-\$ kubectl get deploy -n staging					
NAME READY UP-TO-DATE AVAILABLE AG	ιΕ				
backend-deployment 3/3 3 3 6h	141m				
candidate@node-1:~\$ vim ~/spicy-pikachu/backend-deployme	ent.yaml				
candidate@node-1:~\$ Kubectl config use-context k8s					
Switched to context "K85".	and a deal of a broaders				
candidategnode-1:-5 Kubecti set serviceaccount deploy	app-1 app -n Trontend				
candidateGrade 1.5 kubectl config use context kee					
Switched to context "kBe"					
candidate@node-l:-\$ vim -/orpmpt-escargot/buffalo-deployment.vaml					
candidate@node_listy im _/prompt-escargot/buffalo-deployment.yamc					
candidateGnode_lisS kubect[gn0]v = f -/promot-escargot/buffalo-deployment.vaml					
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 ContainerC	reating 0 8s				
candidate@node-1:-\$ kubectl get deploy -n gorilla					
NAME READY UP-TO-DATE AVAILABLE AG	E				
buffalo-deployment 1/1 1 1 6h	138m				
candidate@node-1:-\$					

File Edit View Terminal Tabs Help	
NAME READY STATUS RES	TARTS AGE
backend-deployment-59d449b99d-cxct6 1/1 Running 0	2 0 s
backend-deployment-59d449b99d-h2zjg 0/1 Running 0	9s
backend-deployment-7897617415-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 Gh41m	
candidate@node-1:~\$ vim -/spicy-pikachu/backend-deployment.	yaml
candidate@node-1:~\$ kubectl config use-context k8s	 Provide
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-deploymen	t.yaml
candidate@node-1:-\$ vim ~/prompt-escargot/buffalo-deploymen	t.yaml
<pre>candidate@node-1:-\$ kubectl apply -f ~/prompt-escargot/bu</pre>	ffalo-deployment.yaml
deployment.apps/buffalo-deployment configured	
<pre>candidate@node-1:-\$ kubectl get pods -n gorilla</pre>	
NAME READY STATUS	RESTARTS AGE
buffalo-deployment-776844df7f-r5fsb 1/1 Running	0 6h38m
<pre>buffalo-deployment-859898c6f5-zx5gj 0/1 ContainerCrea</pre>	ting 0 8s
<pre>candidate@node-1:~\$ kubectl get deploy -n gorilla</pre>	
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".	
<pre>candidate@node-1:~\$ kubectl edit deploy ckad00017-deploymen</pre>	t -n ckad00017

File Edit View Terminal Tabs Help		
Please edit the object below. Lines beginning with a '#' will be ignored, # and an empty file will abort the edit. If an error occurs while saving this file will # reopened with the relevant failures. #	be	
piVersion: apps/vl		
kind: Deployment		
netadata:		
annotations:		
deployment.kubernetes.io/revision: "1"		
creationTimestamp: "2022-09-24T04:27:03Z"		
generation: 1		
labels:		
app: nginx		
name: ckad00017-deployment		
namespace: ckad00017		
resourceversion: "3349"		
uld: 1000/013-Tade-4069-0/41-9429809006/C		
spect second as sound as		
progressuead Lineseconds: 000		
repictory imit. 10		
selector		
matchiabels:		
app: noinx		
strateov:		
rollingUpdate:		
maxSurge: 25%		
maxUnavailable: 25%		
type: RollingUpdate		
template:		
metadata:		
INSERT	19,31	Тор

annual carolisananos P		
File Edit View Terminal Tabs Help		
resourceVersion: "3349" uid: 1cd67613-fade-46e9-b741-94298b9c6e7c		
spec:		
progressDeadlineSeconds: 600		
replicas: 2		
revisionHistoryLimit: 10		
selector:		
matchLabels:		
app: nginx		
strategy:		
rollingUpdate:		
maxSurge: 25%		
maxUnavailable: 25%		
type: RollingUpdate		
template:		
metadata:		
creationTimestamp: null		
labels:		
app: nginx		
spec :		
containers:		
- Image: nginx:Latest		
Imagerul(Policy: Always		
name: ngina		
, containerPort: SA		
protocol: TCP		
resources: {}		
terminationMessagePolicy: File		
dnsPolicy: ClusterFirst		
INSERT	46,14	39%

File Edit View Terminal Tabs Help backend-deployment-59d449b99d-h2zig 0/1 Running 95 backend-deployment-7897617415-b8c85 1/1 6h40m Running 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 6h40m candidate@node-1:-\$ kubectl get deploy -n staging NAME READY UP-TO-DATE AVAILABLE AGE backend-deployment 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 6h38m buffalo-deployment-776844df7f-r5fsb 1/1 Running buffalo-deployment-859898c6f5-zx5gj 0/1 ContainerCreating 0 8s candidate@node-1:~\$ kubectl get deploy -n gorilla NAME READY UP-TO-DATE AVAILABLE AGE 6h38m buffalo-deployment 1/1 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:-\$

File Edit View Terminal Tabs Help		
ouffalo-deployment 1/1 1	1 6h38m	
candidate@node-1:-\$ kubectl config us	e-context k8s	
Switched to context "k8s".		
candidate@node-1:~\$ kubectl edit depl	oy ckad00017-deployment -n cka	d00017
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		이 이 것 같은 것 같은 것 같이 있다. 1911년 2월 2월 2월 2011년 1월 2011년 1
candidate@node-1:-\$ kubectl expose	deploy ckad00017-deployment -n	ckad0001
ckad00014 ckad00015 ckad00017	and a second	
candidate@node-1:~\$ kubectl expose	deploy ckad00017-deployment -n	ckad0001
ckad00014 ckad00015 ckad00017		1. 1000
candidate@node-1:~S kubectl expose	deploy ckaduour-deployment -n	CK909001
CKad00014 CKad00015 CKad00017	dealers also doolog dealers and	-1
chad00014 chad00015 chad00017	deproy ckadeeu/-deproyment -n	ckagaaat
chaudouid chaudouid chaudoui/	daplow chad00017 dealowment	eka40001
ckad00014 ckad00015 ckad00017	deproy ckadebor/-deproyment -n	CK800001
candidate@node_l:s kubert1 evnose	deploy_ckad99917_deploymentp	ckad9991
ckad00014 ckad00015 ckad00017	deptoy cradooor, deptoyment in	CR00001
candidate@node-1:-5 kubectl expose	deploy ckad00017-deployment -n	ckad00017 name=cherry port=8888 type=NodePort
service/cherry exposed	asheed suggester asheed suggest	side and sharing part and spectrum of
candidate@node-1:~\$ kubectl get svc		NUMBER OF THE
NAME TYPE CLUSTER-IP	EXTERNAL-IP PORT(5) AGE	
kubernetes ClusterIP 10.96.0.1	<none> 443/TCP 77d</none>	
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</none>	24s
candidate@node-1:~\$ kubectl get svc		

File Edit View Terminal Tabs Help					
candidate@node-1:~\$ kubectl expose service deploy ckad00017-deployment -n ckad00017name=cherryport=8888type=N					
odePort					
Error from server (NotFound): services "deploy" not found					
Error from server (NotFound): services "ckad00017-deployment" not found					
candidate@node-1:~S kubectl get svc -n ckad00017					
NAME TYPE CLUSTER-IP EXTENNAL-IP PORT(5) AGE					
cherry NodePort 10.100.100.106 <none> 8888:30683/TCP 465</none>					
1 vi vistoriv olivstovi vistoriy					
2 kubarti configuezentari ekse					
2 via via viar					
4 vim -/smicv-pikachu/backend-deplovment.vaml					
5 kubecti apoly -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					
kubecti config use-context k8s					
kubectl set serviceaccount deploy app-1 app -n frontend					
kubectl config use-context k8s					
<pre>vim ~/prompt-escargot/buffalo-deployment.yaml</pre>					
3 kubectl apply -f -/prompt-escargot/buffalo-deployment.yaml					
kubectl get pods -n gorilla					
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 ckad00017name=cherryport=88888type=NodePort					
19 kubectl get svc					
20 Kubecti get svc - n ckadolou/					
21 Rubecti expose service deploy ckadedel/-deployment -n ckadedel/name=cherryport=8888type=NodePort					
22 Kubectriget Svil - n Ckaddoory					
candidate@node-1:~\$					

Answer:

А

Question 4

Refer to Exhibit.



Task:

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

Options:

A- Explanation:

Solution:

e Edit View Terminal Tabs Help	0.205
e programs included with the Ubuntu system are free software; e exact distribution terms for each program are described in the dividual files in /usr/share/doc/*/copyright.	
untu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by plicable law.	
<pre>ndidate@node-1:-\$ vi ~/spicy-pikachu/backend-deployment.yaml ndidate@node-1:-\$ kubectl config use-context sk8s itched to context "sk8s". ndidate@node-1:-\$ vim .vimrc ndidate@node-1:-\$ vim .vimrc ndidate@node-1:-\$ kubectl apply .f ~/spicy-pikachu/backend-deployment.yaml ndidate@node-1:-\$ kubectl apply .f ~/spicy-pikachu/backend-deployment.yaml ployment.apps/backend-deployment configured ndidate@node-1:-\$ kubectl get pods -n staging ME</pre>	
ndidate@node-1:~\$	

Exemplant, candidator another I

. .

А

Question 5

Question Type: MultipleChoice

Refer to Exhibit.



Task

A Deployment named backend-deployment in namespace staging runs a web application on port 8081.

The Deployment's manifest files can be found at ~/spicy-pikachu/backend-deployment.yaml.

Modify the Deployment specifying a readiness probe using path /healthz .

Set initialDelaySeconds to 8 and periodSeconds to 5.

Options:

A- Explanation:

Solution:

File Edit View Terminal Tabs Help

Warning: Permanently added '172.31.17.21' (ECDSA) to the list of known hosts.

The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

candidate@node-1:~\$ vi ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:~\$ kubectl config use-context sk8s
Switched to context "sk8s".
candidate@node-1:~\$ vim .vimrc
candidate@node-1:~\$ vim ~/spicy-pikachu/backend-deployment.yaml

```
File Edit View Terminal Tabs Help
apiVersion: apps/v1
kind: Deployment
metadata:
 name: backend-deployment
  namespace: staging
spec:
  selector:
    matchLabels:
     app: nginx
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
       - name: nginx
          image: nginx:1.14.2
          ports:
            - containerPort: 8081
          readinessProbe:
              initialDelaySeconds: 8
              periodSeconds: 5
              httpGet:
                 path: /healthz
                 port: 8081
          volumeMounts:
            - mountPath: /etc/nginx/conf.d/
              name: config
            - mountPath: /usr/share/nginx/html/
              name: www
 - INSERT --
                                                                                                         26,28
                                                                                                                       Top
```

```
File Edit View Terminal Tabs Help
Warning: Permanently added '172.31.17.21' (ECDSA) to the list of known hosts.
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
candidate@node-1:~$ vi ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:-$ kubectl config use-context sk8s
Switched to context "sk8s".
candidate@node-1:~$ vim .vimrc
candidate@node-1:~$ vim ~/spicy-pikachu/backend-deployment.yaml
candidate@node-1:~$ kubectl apply -f ~/spicy-pikachu/backend-deployment.yaml
deployment.apps/backend-deployment configured
candidate@node-1:~$ kubectl get pods -n staging
NAME
                                      READY STATUS
                                                       RESTARTS
backend-deployment-59d449b99d-cxct6
                                      1/1
                                              Running
                                                       Θ
backend-deployment-59d449b99d-h2zig
                                     0/1
                                              Running
                                                       Θ
backend-deployment-78976f74f5-b8c85
                                     1/1
                                              Running
                                                       Θ
backend-deployment-78976f74f5-flfsj
                                     1/1
                                              Running
                                                       Θ
```

READY UP-TO-DATE

READY UP-TO-DATE

candidate@node-1:~\$ vim ~/spicy-pikachu/backend-deployment.yaml

3

AVAILABLE

AVAILABLE

3

AGE

AGE

6h40m

6h41m

candidate@node-1:-\$ kubectl get deploy -n staging

candidate@node-1:~\$ kubectl get deploy -n staging

3/3

3/3

AGE

20s

6h40m

6h40m

95

NAME

NAME

Answer:

backend-deployment

backend-deployment

Question 6

Question Type: MultipleChoice

Refer to Exhibit.

Set configuration context:					
[student@node-1] \$ k	kubectl				
config use-context k	8s				

Given a container that writes a log file in format A and a container that converts log files from format A to format B, create a deployment that runs both containers such that the log files from the first container are converted by the second container, emitting logs in format B.

Task:

* Create a deployment named deployment-xyz in the default namespace, that:

* Includes a primary

Ifccncf/busybox:1 container, named logger-dev

* includes a sidecar lfccncf/fluentd:v0.12 container, named adapter-zen

* Mounts a shared volume /tmp/log on both containers, which does not persist when the pod is deleted

* Instructs the logger-dev

container to run the command

```
while true; do
echo "i luv cncf" >> /
tmp/log/input.log;
sleep 10;
done
```

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

i luv cncf i luv cncf i luv cncf

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

Options:

A- Explanation:

Solution:



Readme >_ Web Terminal

THELINUX FOUNDATION

<pre>apiVersion: apps/v1 kind: Deployment metadata: creationTimestamp: null labels: app: deployment-xyz spec: replicas: 1 selector: matchLabels: app: deployment-xyz strategy: {} template: metadata: creationTimestamp: null labels: app: deployment-xyz spec: containers: containers: } } </pre>		*
- image: lfccncf/busybox:1		
resources: {}		
status: ()		
"deployment_xyz.yml" 24L, 434C	3,1	All -

Readme >_ Web Terminal

THELINUX FOUNDATION

kind: Deployment		÷
metadata:		
labels:		
app: deployment-xyz		
name: deployment-xyz		
spec:		
replicas: 1		
selector:		
matchLabels:		
app: deployment-xyz		
template:		
metadata:		
labels:		
app: deployment-xyz		
spec:		
volumes:		
- name: myvol1		
emptyDir: ()		
containers:		
- image: lfccncf/busybox:1		
name: logger-dev		
volumeMounts:		
- name: myvol1		
mountPath: /tmp/log		
- image: lfccncf/fluentd:v0.12		
name: adapter-zen		
3 lines yanked	27,22	Bot 👻

```
replicas: 1
selector:
matchLabels:
    app: deployment-xyz
template:
    metadata:
    labels:
    app: deployment-xyz
spec:
    volumes:
    - name: myvol1
    emptyDir: {}
    containers:
    - image: lfcencf/busybox:1
    name: logger-dev
    command: ["/bin/sh","-c","while [ true ]; do echo 'i luv cnef' >> /tmp/log/input.log; al
eep 10; done"]
    volumeMounts:
    - name: myvol1
    mountPath: /tmp/log
    image: lfcencf/fluentd:v0.12
    name: adapter=zen
    command: ["/bin/sh","-c","tail -f /tmp/log/input.log >> /tmp/log/output.log"]
    volumeMounts:
    - name: myvol1
    mountPath: /tmp/log
    29,83
    Bot
```

🛱 Readme 🔪 Web Terminal

metadata:		
labels:		
app: deployment-xyz		
spec:		
volumes:		
- name: myvol1		
emptyDir: ()		
- name: myvol2		
configMap:		
name: logconf		
containers:		
- image: lfccncf/busybox:1		
name: logger-dev		
		.log; sl
volumeMounts:		
- name: myvol1		
mountPath: /tmp/log		
- image: lfccncf/fluentd:v0.12		
name: adapter-zen		
command: ["/bin/sh","-c","tail -f /tmp/log/input.log >> /tmp/log/		
volumeMounts:		
- name: myvol1		
mountPath: /tmp/log		
name: myvol2		
mountPath: /Iluentd/etc	37 33	Pat
	37,33	BOL

>_ Web Terminal

🕮 Readme

	-			
student@node-1:~\$	kubect	l create -f de	eployment_xy	z.yml
deployment.apps/c	leploymen	nt-xyz created	1	
student@node-1:~	kubect!	l get deployme	ent	
NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment-xyz	0/1	1	0	5 3
student@node-1:~	kubect!	l get deployme	ent	
NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment-xyz	0/1	1	0	95
student@node-1:~{	kubect!	l get deployme	ent	
NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment-xyz	1/1	1	1	12s
student@node-1:~{				

Answer:

А

Question 7

Question Type: MultipleChoice

Refer to Exhibit.



Context

A project that you are working on has a requirement for persistent data to be available.

Task

To facilitate this, perform the following tasks:

* Create a file on node sk8s-node-0 at /opt/KDSP00101/data/index.html with the content Acct=Finance

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

* Create a PefsissentVolumeClaim named task-pv-claim that requests a volume of at least 100Mi and specifies an access mode of ReadWriteOnce

* Create a pod that uses the PersistentVolmeClaim as a volume with a label app: my-storage-app mounting the resulting volume to a mountPath /usr/share/nginx/html inside the pod



Options:

A- Explanation: Solution:

🕮 Readme 🔪 Web Terminal

THE LINUX FOUNDATION

student@node-1:~\$ kubectl config use-context sk8s Switched to context "sk8s". student@node-1:~\$

🕮 Readme 🔪 Web Terminal

THELINUX FOUNDATION

* Documentation: https://help.ubuntu.com * Management: https://landscape.canonical.com * Support: https://ubuntu.com/advantage System information as of Fri Oct 9 08:52:09 UTC 2020

System load:	2.02	Users logged in:	0
Usage of /:	10.3% of 242.29GB	IP address for eth0:	10.250.3.115
Memory usage:	2%	IP address for docker0:	172.17.0.1
Swap usage:	0%	IP address for cni0:	10.244.1.1
Processes:	38		

* Kubernetes 1.19 is out! Get it in one command with:

sudo snap install microk8s --channel=1.19 --classic

https://microk8s.io/ has docs and details.

7 packages can be updated. 1 update is a security update.

New release '20.04.1 LTS' available. Run 'do-release-upgrade' to upgrade to it.

student@sk8s-node-0:~\$



student@sk8s-node-0:~\$ echo 'Acct=Finance' > /opt/KDSP00101/data/index.html
student@sk8s-node-0:~\$ vim pv.yml

💷 Readme	▶_ Web Terminal		
~			÷
INSERT		0,1 All	r .

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

>_ Web Terminal

💷 Readme

THELINUX FOUNDATION

🕮 Readme 👌 🔪 Web Terminal

student@sk8s-no	de-0:~\$	kubectl create -f	pv.yml					
persistentvolum	e/task-p	v-volume created						
student@sk8s-no	de-0:~\$	kubectl create -f	pvc.yml					
persistentvolum	eclaim/t	ask-pv-claim creat	ed					
student@sk8s-no	de-0:~\$	kubectl get pv						
NAME	CAPACI	TY ACCESS MODES	RECLAIM	POLICY	STATUS	CLAIM		STO
RAGECLASS REA	SON AG	E						
task-pv-volume	1Gi	RWO	Retain		Bound	default/task-	-pv-claim	sto
rage	11	8						
student@sk8s-no	de-0:~\$	kubectl get pvc						
NAME	STATUS	VOLUME	CAPACITY	ACCESS	MODES	STORAGECLASS	AGE	
task-pv-claim	Bound	task-pv-volume	1Gi	RWO		storage	9s	
student@sk8s-no	de-0:~\$	vim pod.yml						

apiVersion: v1		
kind: Pod		
metadata:		
name: mypod		
labels:		
app: my-storage-app		
apec:		
containers:		
- name: myfrontend		
image: nginx		
volumeMounts:		
- mountPath: "/usr/share/nginx/html."		
name: mypod		
volumes:		
- name: mypod		
persistentVolumeClaim:		
claimName: task-pv-claim		
	17,32	A11 -

student@sk8s-node-0:~\$ kubectl create -f pod.yml
pod/mypod created
student@sk8s-node-0:~\$ kubectl get

Readme >_ Web Terminal

student@sk8s-node-0:~\$ kubectl get pods READY STATUS RESTARTS NAME AGE 0/1 ContainerCreating mypod 0 43 student@sk8s-node-0:~\$ kubect1 get pods NAME READY STATUS RESTARTS AGE 0/1 ContainerCreating 0 83 mypod student@sk8s-node-0:~\$ kubect1 get pods NAME READY STATUS RESTARTS AGE mypod 1/1 Running 0 10s student@sk8s-node-0:~\$ logout Connection to 10.250.3.115 closed. student@node-1:~\$

🕮 Readme 👌 🔪 Web Terminal

Answer:

А

Question 8

Question Type: MultipleChoice

Refer to Exhibit.



Set Configuration Context:

[student@node-1] \$ | kubectl

Config use-context k8s

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.



Options:

A- Explanation:

To find the broken pod and store its name and namespace to /opt/KDOB00401/broken.txt, you can use the kubectl get pods command and filter the output by the status of the pod.

kubectl get pods --field-selector=status.phase=Failed -o jsonpath='{.items[*].metadata.namespace}/{.items[*].metadata.name}' > /opt/KDOB00401/broken.txt

This command will list all pods with a status of Failed and output their names and namespaces in the format <namespace>/. The output is then written to the /opt/KDOB00401/broken.txt file.

To store the associated error events to a file /opt/KDOB00401/error.txt, you can use the kubectl describe command to retrieve detailed information about the pod, and the grep command to filter the output for error events.

kubectl describe pods --namespace | grep -i error -B5 -A5 > /opt/KDOB00401/error.txt

Replace and with the name and namespace of the broken pod you found in the previous step.

This command will output detailed information about the pod, including error events. The grep command filters the output for lines containing 'error' and also prints 5 lines before and after the match.

To fix the issue, you need to analyze the error events and find the root cause of the issue.

It could be that the application inside the pod is not running, the container image is not available, the pod has not enough resources, or the liveness probe configuration is incorrect.

Once you have identified the cause, you can take appropriate action, such as restarting the application, updating the container image, increasing the resources, or modifying the liveness probe configuration.

After fixing the issue, you can use the kubectl get pods command to check the status of the pod and ensure

Answer:

Question 9

Question Type: MultipleChoice

Refer to Exhibit.

terre in G eneral terre	
[student@node-1] \$ use-context nk8s	kubectl config

Set Configuration Context:

[student@node-1] \$ | kubectl

Config use-context k8s

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.

Options:

A- Explanation:

To allow a pod to send and receive traffic only to and from specific pods, you can use network policies in Kubernetes.

First, you will need to create a network policy that defines the allowed traffic. You can create a network policy yaml file with the following rules:

apiVersion: networking.k8s.io/v1

kind: NetworkPolicy

metadata:

name: newpod-network-policy

namespace: default

spec:

podSelector:

matchLabels:

app: kdsn00201-newpod

ingress:

- from:

- podSelector:

matchLabels:

app: web

- podSelector:

matchLabels:

app: storage

This policy will only allow incoming traffic to the pod with the label app=kdsn00201-newpod from pods with the label app=web or app=storage. If you have different labels on your web and storage pods please update the matchLabels accordingly. Once you have created the network policy, you can apply it to the cluster by running the following command:

kubectl apply -f <network-policy-file>.yaml

This will apply the network policy to the cluster, and the newpod will only be able to send and receive traffic to and from the web and storage pods.

Please note that, NetworkPolicy resource is not available by default, you need to enable the NetworkPolicy feature on your Kubernetes cluster. This feature is enabled by default on some clusters and must be explicitly enabled on others. You can check if NetworkPolicy is available by running the command kubectl api-versions | grep networking

Also, you need to ensure that the pods that you want to allow traffic to and from are running on the same namespace.

Answer:

А

To Get Premium Files for CKAD Visit

https://www.p2pexams.com/products/ckad

For More Free Questions Visit

https://www.p2pexams.com/linux-foundation/pdf/ckad

