

LinuxFoundation.CKA.v2024-07-08.q69

Exam Code:	CKA
Exam Name:	Certified Kubernetes Administrator (CKA) Program Exam
Certification Provider:	Linux Foundation
Free Question Number:	69
Version:	v2024-07-08
# of views:	1838
# of Questions views:	690
https://www.freepdfdumps.com/LinuxFoundation.CKA.v2024-07-08.q69.html	

NEW QUESTION: 1

List pod logs named "frontend" and search for the pattern "started" and write it to a file "/opt/error-logs"

Answer:

See the solution below.

Explanation

```
Kubectl logs frontend | grep -i "started" > /opt/error-logs
```

NEW QUESTION: 2

Create a deployment as follows:

Name: nginx-app

Using container nginx with version 1.11.10-alpine

The deployment should contain

Next, deploy the application with new version , by performing a rolling update.

Finally, rollback that update to the previous version

Answer:

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\7 B.JPG



F:\Work\Data Entry Work\Data Entry\20200827\CKA\7 C.JPG



F:\Work\Data Entry Work\Data Entry\20200827\CKA\7 D.JPG

```
Readme > Web Terminal THE LINUX FOUNDATION  
root@node-1:~# k create deploy nginx-app --image=nginx:1.11.10-alpine --dry-run=client -o y  
aml > app.yaml  
root@node-1:~# vim app.yaml  
root@node-1:~# k create -f app.yaml  
deployment.apps/nginx-app created  
root@node-1:~#  
root@node-1:~#  
root@node-1:~# k set image deploy nginx-app nginx=nginx:1.11.13-alpine --record  
deployment.apps/nginx-app image updated  
root@node-1:~# k rollout undo deploy nginx-app  
deployment.apps/nginx-app rolled back  
root@node-1:~#
```

NEW QUESTION: 3

Score: 4%



Task

Create a persistent volume with name app-data , of capacity 1Gi and access mode ReadOnlyMany. The type of volume is hostPath and its location is /srv/app-data .

Answer:

Solution:

```
#vi pv.yaml
```

```
apiVersion: v1
```

```
kind: PersistentVolume
```

```
metadata:
```

```
name: app-config
spec:
capacity:
storage: 1Gi
accessModes:
- ReadOnlyMany
hostPath:
path: /srv/app-config
#
kubectl create -f pv.yaml
```

NEW QUESTION: 4

List the nginx pod with custom columns POD_NAME and POD_STATUS

Answer:

```
kubectl get po -o=custom-columns="POD_NAME:.metadata.name,
POD_STATUS:.status.containerStatuses[].state"
```

NEW QUESTION: 5

Create a pod that echo "hello world" and then exists. Have the pod deleted automatically when it's completed

Answer:

```
kubectl run busybox --image=busybox -it --rm --restart=Never --
/bin/sh -c 'echo hello world'
kubectl get po # You shouldn't see pod with the name "busybox"
```

NEW QUESTION: 6

Resume the rollout of the deployment

Answer:

```
kubectl rollout resume deploy webapp
```

NEW QUESTION: 7

Set the node named ek8s-node-1 as unavailable and reschedule all the pods running on it.

Answer:

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\19 B.JPG

```
root@node-1:~# kubectl config use-context ek8s
Switched to context "ek8s".
root@node-1:~# k drain ek8s-node-1 --ignore-daemonsets --delete-local-data --force
node/ek8s-node-1 cordoned
WARNING: ignoring DaemonSet-managed Pods: kube-system/kube-flannel-ds-amd64-qj7w8, kube-syst
em/kube-proxy-x7xkv
evicting pod default/nginx-568f5649b8-c9zkj
evicting pod kube-system/metrics-server-64b57fd654-cktk5
```

NEW QUESTION: 8

A Kubernetes worker node, named wk8s-node-0 is in state NotReady. Investigate why this is the case, and perform any appropriate steps to bring the node to a state, ensuring that any changes are made permanent.

You can ssh to the failed node using:

```
[student@node-1] $ | ssh Wk8s-node-0
```

You can assume elevated privileges on the node with the following command:

```
[student@w8ks-node-0] $ | sudo -i
```

Answer:

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\20 C.JPG

```
Readme >_ Web Terminal THE LINUX FOUNDATION

root@node-1:~# kubectl config use-context wk8s
Switched to context "wk8s".
root@node-1:~# k get nodes
NAME          STATUS    ROLES    AGE   VERSION
wk8s-master-0 Ready     master   77d   v1.18.2
wk8s-node-0   NotReady <none>   77d   v1.18.2
wk8s-node-1   Ready     <none>   77d   v1.18.2
root@node-1:~# ssh wk8s-node-0
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\20 D.JPG

```
Readme >_ Web Terminal THE LINUX FOUNDATION

wk8s-node-0   NotReady <none>   77d   v1.18.2
wk8s-node-1   Ready     <none>   77d   v1.18.2
root@node-1:~# ssh wk8s-node-0
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1109-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Are you ready for Kubernetes 1.19? It's nearly here. Try RC3 with
   sudo snap install microk8s --channel=1.19/candidate --classic
   https://microk8s.io/ has docs and details

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

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

student@wk8s-node-0:~$ sudo -i
root@wk8s-node-0:~# systemctl restart kubelet
root@wk8s-node-0:~# systemctl enable kubelet
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\20 E.JPG

```
Readme Web Terminal THE LINUX FOUNDATION
https://microk8s.io/ has docs and details.

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

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

student@wk8s-node-0:~$ sudo -i
root@wk8s-node-0:~# systemctl restart kubelet
root@wk8s-node-0:~# systemctl enable kubelet
Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to /lib/systemd/system/kubelet.service.
root@wk8s-node-0:~# exit
logout
student@wk8s-node-0:~$ exit
logout
Connection to 10.250.64.11 closed.
root@node-1:~# k get nodes
NAME                STATUS    ROLES    AGE   VERSION
wk8s-master-0      Ready    master   77d   v1.18.2
wk8s-node-0        Ready    <none>   77d   v1.18.2
wk8s-node-1        Ready    <none>   77d   v1.18.2
root@node-1:~#
```

NEW QUESTION: 9

Set CPU and memory requests and limits for existing pod name "nginx-prod".

Set requests for CPU and Memory as 100m and 256Mi respectively

Set limits for CPU and Memory as 200m and 512Mi respectively

A. kubectl get po

kubectl set resources po nginx-prod --

limits=cpu=200m,memory=512Mi --requests=cpu=100m,memory=256Mi

//Verify

kubectl top po

kubectl describe po nginx-prod

B. kubectl get po

kubectl set resources po nginx-prod --

limits=cpu=200m,memory=512Mi --requests=cpu=100m,memory=256Mi

//Verify

kubectl describe po nginx-prod

Answer: (SHOW ANSWER)

NEW QUESTION: 10

Score: 7%



Task

Create a new nginx Ingress resource as follows:

- * Name: ping
- * Namespace: ing-internal
- * Exposing service hi on path /hi using service port 5678



Answer:

See the solution below.

Explanation

Solution:

```
vi ingress.yaml
```

```
#
```

```
apiVersion: networking.k8s.io/v1
```

```
kind: Ingress
```

```
metadata:
```

```
name: ping
```

```
namespace: ing-internal
```

```
spec:
```

```
rules:
```

```
- http:
```

paths:

- path: /hi

pathType: Prefix

backend:

service:

name: hi

port:

number: 5678

#

kubectl create -f ingress.yaml

NEW QUESTION: 11

Create a pod named kucc8 with a single app container for each of the following images running inside (there may be between 1 and 4 images specified):
nginx + redis + memcached.

Answer:

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\5 B.JPG



```
root@node-1:~# vim ds.yaml
iroot@node-1:~# k create -f ds.yaml
daemonset.apps/ds-kusc00201 created
root@node-1:~# k get ds
NAME          DESIRED  CURRENT  READY  UP-TO-DATE  AVAILABLE  NODE SELECTOR  AGE
ds-kusc00201  2        2        2      2           2          <none>         4s
root@node-1:~# vim /opt/KUCC00108/pod-spec-KUCC00108.yaml
root@node-1:~# k create -f /opt/KUCC00108/pod-spec-KUCC00108.yaml
pod/hungry-bear created
root@node-1:~# k get po
NAME          READY   STATUS    RESTARTS   AGE
cpu-utilizer-98b9se  1/1    Running   0          5h50m
cpu-utilizer-ab2d3s  1/1    Running   0          5h50m
cpu-utilizer-kipb9a  1/1    Running   0          5h50m
ds-kusc00201-2r2k9  1/1    Running   0          4m50s
ds-kusc00201-hzm9q  1/1    Running   0          4m50s
foo            1/1    Running   0          5h52m
front-end      1/1    Running   0          5h52m
hungry-bear    1/1    Running   0          42s
webserver-84c55967f4-qzjcv  1/1    Running   0          6h7m
webserver-84c55967f4-t0ff  1/1    Running   0          6h7m
root@node-1:~# k run nginx --image=nginx --dry-run=client -o yaml > nginx.yaml
root@node-1:~# vim nginx.yaml
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\5 C.JPG

Create a pod with environment variables as var1=value1. Check the environment variable in pod

A. `kubectl run nginx --image=nginx --restart=Never --env=var1=value1`

then

`kubectl exec -it nginx -- env`

or

`kubectl describe po nginx | grep value1`

B. `kubectl run nginx --image=nginx --restart=Never --env=var1=value1`

then

`kubectl exec -it nginx -- env`

or

`kubectl exec -it nginx -- sh -c 'echo $var1'`

or

`kubectl describe po nginx | grep value1`

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 13

Create a Kubernetes secret as follows:

* Name: super-secret

* password: bob

Create a pod named pod-secrets-via-file Image, which mounts a secret named super-secret at /secrets.

Create a second pod named pod-secrets-via-env Image, which exports password as

CONFIDENTIAL

Answer:

See the solution below.

Explanation

solution


```

root@node-1:~# k create -f secret.yaml
pod/pod-secrets-via-file created
root@node-1:~# vim secret1.yaml
root@node-1:~# k create -f secret1.yaml
pod/pod-secrets-via-env created
root@node-1:~# k get po
NAME                                READY   STATUS    RESTARTS   AGE
cpu-utilizer-98b9se                 1/1     Running   0           6h25m
cpu-utilizer-ab2d3s                 1/1     Running   0           6h25m
cpu-utilizer-kipb9a                 1/1     Running   0           6h25m
ds-kusc00201-2r2k9                 1/1     Running   0           40m
ds-kusc00201-hzm9q                 1/1     Running   0           40m
foo                                  1/1     Running   0           6h28m
front-end                           1/1     Running   0           6h27m
hungry-bear                         1/1     Running   0           36m
kucc8                                3/3     Running   0           34m
nginx-app-848cfcf495-9prjh          1/1     Running   0           19m
nginx-app-848cfcf495-q12kh          1/1     Running   0           19m
nginx-app-848cfcf495-rjz68         1/1     Running   0           19m
nginx-kusc00101                     1/1     Running   0           26m
pod-secrets-via-env                 1/1     Running   0           4s
pod-secrets-via-file               1/1     Running   0           106s
webserver-84c55967f4-qzjcv         1/1     Running   0           6h43m
webserver-84c55967f4-t479l        1/1     Running   0           6h43m
root@node-1:~#

```

NEW QUESTION: 14

Print pod name and start time to "/opt/pod-status" file

Answer:

```
kubectl get pods -o=jsonpath='{range .items[*]}{.metadata.name}{"\t"}{.status.podIP}{"\n"}{end}'
```

NEW QUESTION: 15

Get list of all the pods showing name and namespace with a jsonpath expression.

Answer:

See the solution below.

Explanation

```
kubectl get pods -o=jsonpath="{.items[*]['metadata.name']
, 'metadata.namespace']}"
```

NEW QUESTION: 16

Configure the kubelet systemd-managed service, on the node labelled with name=wk8s-node-1, to launch a pod containing a single container of Image automatically. Any spec files required should be placed in the /etc/kubernetes/manifests You can ssh to the appropriate node using:

```
[student@node-1] $ ssh wk8s-node-1
```

You can assume elevated privileges on the node with the following command:

```
[student@wk8s-node-1] $ |sudo -i
```

Answer:

See the solution below.

Explanation

solution

```
Readme >_ Web Terminal THE LINUX FOUNDATION

root@node-1:~#
root@node-1:~# kubectl config use-context wk8s
Switched to context "wk8s".
root@node-1:~# ssh wk8s-node-1
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1109-aws x86_64)

Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

* Are you ready for Kubernetes 1.19? It's nearly here! Try RC3 with
sudo snap install microk8s --channel=1.19/candidate --classic

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

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

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

student@wk8s-node-1:~$ sudo -i
root@wk8s-node-1:~# vim /var/lib/kubelet/config.yaml
```

```
Readme >_ Web Terminal THE LINUX FOUNDATION

clientCAFile: /etc/kubernetes/pki/ca.crt
authorization:
  mode: Webhook
  webhook:
    cacheAuthorizedTTL: 0s
    cacheUnauthorizedTTL: 0s
clusterDNS:
- 10.96.0.10
clusterDomain: cluster.local
cpuManagerReconcilePeriod: 0s
evictionPressureTransitionPeriod: 0s
fileCheckFrequency: 0s
healthzBindAddress: 127.0.0.1
healthzPort: 10248
httpCheckFrequency: 0s
imageMinimumGCAge: 0s
kind: KubeletConfiguration
nodeStatusReportFrequency: 0s
nodeStatusUpdateFrequency: 0s
rotateCertificates: true
runtimeRequestTimeout: 0s
staticPodPath: /etc/kubernetes/manifests
streamingConnectionIdleTimeout: 0s
syncFrequency: 0s
:W
```

```
root@node-1:~# ssh wk8s-node-1
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1109-aws x86_64)
```

- * Documentation: <https://help.ubuntu.com>
- * Management: <https://landscape.canonical.com>
- * Support: <https://ubuntu.com/advantage>

* Are you ready for Kubernetes 1.19? It's nearly here! Try RC3 with `sudo snap install microk8s --channel=1.19/candidate --classic`

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

```
4 packages can be updated.
1 update is a security update.
```

```
New release '16.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

```
student@wk8s-node-1:~$ sudo -i
root@wk8s-node-1:~# vim /var/lib/kubelet/config.yaml
root@wk8s-node-1:~# cd /etc/kubernetes/manifests
root@wk8s-node-1:/etc/kubernetes/manifests#
root@wk8s-node-1:/etc/kubernetes/manifests# vim pod.yaml
```



```
apiVersion: v1
kind: Pod
metadata:
  name: webtool
spec:
  containers:
  - name: webtool
    image: httpd
```



```
Readme > Web Terminal THE LINUX FOUNDATION

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

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

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

student@wk8s-node-1:~$ sudo -i
root@wk8s-node-1:~# vim /var/lib/kubelet/config.yaml
root@wk8s-node-1:~# cd /etc/kubernetes/manifests
root@wk8s-node-1:/etc/kubernetes/manifests# vim pod.yaml
root@wk8s-node-1:/etc/kubernetes/manifests# systemctl restart kubelet
root@wk8s-node-1:/etc/kubernetes/manifests# systemctl enable kubelet
root@wk8s-node-1:/etc/kubernetes/manifests# exit
logout
student@wk8s-node-1:~$ exit
logout
Connection to 10.250.5.39 closed.
root@node-1:~# k get po
NAME                READY   STATUS    RESTARTS   AGE
webtool-wk8s-node-1  1/1     Running   0           11s
root@node-1:~#
```

Valid CKA Dumps shared by Actual4test.com for Helping Passing CKA Exam!

Actual4test.com now offer the **newest CKA exam dumps**, the Actual4test.com CKA exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com CKA dumps with Test Engine here:

https://www.actual4test.com/CKA_examcollection.html (85 Q&As Dumps, **30%OFF** Special Discount: **Freepdfdumps**)

NEW QUESTION: 17

A Kubernetes worker node, named . Investigate why this is the case, and perform any appropriate steps to bring the node to a state, ensuring that any changes are made permanent.

You can ssh to the failed node using:

```
[student@node-1] $ | ssh Wk8s-node-0
```

You can assume elevated privileges on the node with the following command:

```
[student@w8ks-node-0] $ | sudo -i
```

Answer:

See the solution below.

Explanation

solution

```
root@node-1:~# kubectl config use-context wk8s
Switched to context "wk8s".
root@node-1:~# k get nodes
NAME                STATUS    ROLES    AGE   VERSION
wk8s-master-0      Ready     master   77d   v1.18.2
wk8s-node-0        NotReady <none>   77d   v1.18.2
wk8s-node-1        Ready     <none>   77d   v1.18.2
root@node-1:~# ssh wk8s-node-0
```

```
wk8s-node-0      NotReady <none>   77d   v1.18.2
wk8s-node-1      Ready     <none>   77d   v1.18.2
root@node-1:~# ssh wk8s-node-0
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1109-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Are you ready for Kubernetes 1.19? It's nearly here. Try RC3 with
   sudo snap install microk8s --channel=1.19/candidate --classic
   https://microk8s.io/ has docs and details.

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

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

student@wk8s-node-0:~$ sudo -i
root@wk8s-node-0:~# systemctl restart kubelet
root@wk8s-node-0:~# systemctl enable kubelet
```

```
Readme > Web Terminal THE LINUX FOUNDATION
https://microk8s.io/ has docs and details. FOUNDATION

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

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

student@wk8s-node-0:~$ sudo -i
root@wk8s-node-0:~# systemctl restart kubelet
root@wk8s-node-0:~# systemctl enable kubelet
Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to /lib/systemd/system/kubelet.service.
root@wk8s-node-0:~# exit
logout
student@wk8s-node-0:~$ exit
logout
Connection to 10.250.5.34 closed.
root@node-1:~# k get nodes
NAME                STATUS    ROLES    AGE   VERSION
wk8s-master-0      Ready    master   77d   v1.18.2
wk8s-node-0        Ready    <none>   77d   v1.18.2
wk8s-node-1        Ready    <none>   77d   v1.18.2
root@node-1:~#
```

NEW QUESTION: 18

Create a namespace called 'development' and a pod with image nginx called nginx on this namespace.

Answer:

```
kubectl create namespace development
kubectl run nginx --image=nginx --restart=Never -n development
```

NEW QUESTION: 19

Create an nginx pod and set an env value as 'var1=val1'. Check the env value existence within the pod

A. `kubectl run nginx --image=nginx --restart=Never --env=var1=val1`

then

```
kubectl exec -it nginx -- env
```

or

```
kubectl exec -it nginx -- sh -c 'echo $var1'
```

or

```
kubectl describe po nginx | grep val1
```

or

```
kubectl run nginx --restart=Never --image=nginx --env=var1=val1
-it --rm - env
```

B. `kubectl run nginx --image=nginx --restart=Never --env=var1=val1`

then

kubectl exec -it nginx -- env

or

kubectl run nginx --restart=Never --image=nginx --env=var1=val1

-it --rm -- env

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 20

Check to see how many worker nodes are ready (not including nodes tainted NoSchedule) and write the number to /opt/KUCC00104/kucc00104.txt.

Answer:

solution

```
root@node-1:~# k scale deploy webserver --replicas=6
deployment.apps/webserver scaled
root@node-1:~# k get deploy
NAME          READY  UP-TO-DATE  AVAILABLE  AGE
nginx-app     3/3    3            3           29m
webserver     6/6    6            6           6h50m
root@node-1:~#
root@node-1:~# k get nodes
NAME           STATUS  ROLES    AGE   VERSION
k8s-master-0  Ready   master   77d   v1.18.2
k8s-node-0    Ready   <none>   77d   v1.18.2
k8s-node-1    Ready   <none>   77d   v1.18.2
root@node-1:~# vi /opt/KUCC00104/kucc00104.txt
```



NEW QUESTION: 21

For this item, you will have to ssh to the nodes ik8s-master-0 and ik8s-node-0 and complete all tasks on these nodes. Ensure that you return to the base node (hostname: node-1) when you have completed this item.

Context

As an administrator of a small development team, you have been asked to set up a Kubernetes cluster to test the viability of a new application.

Task

You must use kubeadm to perform this task. Any kubeadm invocations will require the use of the `--ignore-preflight-errors=all` option.

- * Configure the node ik8s-master-0 as a master node. .
- * Join the node ik8s-node-0 to the cluster.

Answer:

See the solution below.

Explanation

solution

You must use the kubeadm configuration file located at `/etc/kubeadm.conf` when initializing your cluster.

You may use any CNI plugin to complete this task, but if you don't have your favourite CNI plugin's manifest URL at hand, Calico is one popular option:

<https://docs.projectcalico.org/v3.14/manifests/calico.yaml> Docker is already installed on both nodes and has been configured so that you can install the required tools.

NEW QUESTION: 22

An administrator accidentally closed the commit window/screen before the commit was finished. Which two options could the administrator use to verify the progress or success of that commit task? (Choose two)

- A. Traffic Logs
- B. System Logs
- C. Configuration Logs
- D. Task Manager

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 23

Create a deployment spec file that will:

Launch 7 replicas of the nginx Image with the labelapp_runtime_stage=dev deployment name: kual00201 Save a copy of this spec file to /opt/KUAL00201/spec_deployment.yaml (or /opt/KUAL00201/spec_deployment.json).

When you are done, clean up (delete) any new Kubernetes API object that you produced during this task.

Answer:

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\10 B.JPG

```
Readme Web Terminal THE LINUX FOUNDATION
root@node-1:~# k create deploy kual00201 --image=nginx --dry-run=client -o yaml > /opt/KUAL00201/spec_deployment.yaml
root@node-1:~# vim /opt/KUAL00201/spec_deployment.yaml
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\10 C.JPG

```
Readme Web Terminal THE LINUX FOUNDATION
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app_runtime_stage: dev
  name: kual00201
spec:
  replicas: 7
  selector:
    matchLabels:
      app_runtime_stage: dev
  template:
    metadata:
      labels:
        app_runtime_stage: dev
    spec:
      containers:
      - image: nginx
        name: nginx
~
~
~
"/opt/KUAL00201/spec_deployment.yaml" 19L, 320C written
```

NEW QUESTION: 24

Create and configure the service front-end-service so it's accessible through NodePort and routes to the existing pod named front-end.

Answer:

See the solution below.

Explanation

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\8 B.JPG



```
root@node-1:~# k expose po
error: resource(s) were provided, but no name, label selector, or --all flag specified
See 'kubectl expose -h' for help and examples
root@node-1:~# k expose po front-end --name=front-end-service --port=80 --target-port=80 --t
ype=NodePort
Error from server (NotFound): pods "front-end" not found
root@node-1:~# k expose po front-end --name=front-end-service --port=80 --target-port=80 --
type=NodePort
service/front-end-service exposed
root@node-1:~# k get svc
NAME                TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
front-end-service   NodePort      10.103.221.217  <none>           80:31828/TCP     3s
kubernetes          ClusterIP     10.96.0.1        <none>           443/TCP          77d
```

NEW QUESTION: 25

Create a deployment as follows:

Name: nginx-app

Using container nginx with version 1.11.10-alpine

The deployment should contain 3 replicas

Next, deploy the application with new version 1.11.13-alpine, by performing a rolling update.

Finally, rollback that update to the previous version 1.11.10-alpine.

Answer:

solution


```
root@node-1:~# k create deploy nginx-app --image=nginx:1.11.10-alpine --dry-run=client -o y
aml > app.yaml
root@node-1:~# vim app.yaml
root@node-1:~# k create -f app.yaml
deployment.apps/nginx-app created
root@node-1:~#
root@node-1:~#
root@node-1:~# k set image deploy nginx-app nginx=nginx:1.11.13-alpine --record
deployment.apps/nginx-app image updated
root@node-1:~# k rollout undo deploy nginx-app
deployment.apps/nginx-app rolled back
root@node-1:~#
```

NEW QUESTION: 26

Undo the deployment with the previous version and verify everything is Ok

Answer:

kubectl rollout undo deploy webapp kubectl rollout status deploy webapp kubectl get pods

NEW QUESTION: 27

Create a deployment named "myapp" that having 2 replicas with nginx image and expose deployment as service named "myservice"

A. // Create a YAML Template

```
kubectl create deploy myapp --image=nginx --dry-run -o yaml >
myapp.yaml
```

//Update replicas=2 in myapp.yaml file

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
labels:
```

```
app: myapp
```

```
name: myapp
```

```
spec:
```

```
replicas: 2
```

```
selector:
matchLabels:
app: myapp
template:
metadata:
labels:
app: myapp
spec:
containers:
- image: nginx
name: nginx
// Create deployment
kubectl create -f myapp.yaml
// Creating YAML template for service
kubectl expose deployment myapp --type=ClusterIP --port=60 --
target-port=60 --name=myservice --dry-run -o yaml >
myservice.yaml
YAML File:
apiVersion: v1
kind: Service
metadata:
labels:
app: myapp
name: myservice
spec:
ports:
- port: 60
protocol: TCP
targetPort: 80
selector:
app: myapp
type: ClusterIP
kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S)
AGE
kubernetes ClusterIP 10.2.0.1 <none> 443/TCP
158d
myservice ClusterIP 10.2.96.175 <none> 80/TCP
40s
B. // Create a YAML Template
kubectl create deploy myapp --image=nginx --dry-run -o yaml >
```

```
myapp.yaml
//Update replicas=2 in myapp.yaml file
apiVersion: apps/v1
kind: Deployment
metadata:
labels:
app: myapp
name: myapp
spec:
replicas: 2
selector:
matchLabels:
app: myapp
template:
metadata:
labels:
app: myapp
spec:
containers:
- image: nginx
name: nginx
// Create deployment
kubectl create -f myapp.yaml
// Creating YAML template for service
kubectl expose deployment myapp --type=ClusterIP --port=80 --
target-port=80 --name=myservice --dry-run -o yaml >
myservice.yaml
YAML File:
apiVersion: v1
kind: Service
metadata:
labels:
app: myapp
name: myservice
spec:
ports:
- port: 80
protocol: TCP
targetPort: 80
selector:
app: myapp
```

```
type: ClusterIP
kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S)
AGE
kubernetes ClusterIP 10.2.0.1 <none> 443/TCP
158d
myservice ClusterIP 10.2.96.175 <none> 80/TCP
40s
```

Answer: B (LEAVE A REPLY)

NEW QUESTION: 28

Get list of all the nodes with labels

Answer:

```
kubectl get nodes --show-labels
```

NEW QUESTION: 29

Score: 4%



Task

Create a persistent volume with name app-data , of capacity 1Gi and access mode ReadOnlyMany. The type of volume is hostPath and its location is /srv/app-data .

Answer:

See the solution below.

Explanation

Solution:

```
#vi pv.yaml
```

```
apiVersion: v1
```

```
kind: PersistentVolume
```

```
metadata:
```

```
name: app-config
```

```
spec:
```

```
capacity:
```

```
storage: 1Gi
```

```
accessModes:
```

```
- ReadOnlyMany
hostPath:
path: /srv/app-config
#
kubectl create -f pv.yaml
```

NEW QUESTION: 30

How can an administrator configure the NGFW to automatically quarantine a device using Global Protect?

- A. There is no native auto-quarantine feature so a custom script would need to be leveraged.
- B. by adding the device's Host ID to a quarantine list and configure GlobalProtect to prevent users from connecting to the GlobalProtect gateway from a quarantined device.
- C. by exporting the list of quarantined devices to a pdf or csv file by selecting PDF/CSV at the bottom of the Device Quarantine page and leveraging the appropriate XSOAR playbook.
- D. by using security policies, log forwarding profiles, and log settings.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 31

Check the history of deployment

Answer:

```
kubectl rollout history deployment webapp
```

Valid CKA Dumps shared by Actual4test.com for Helping Passing CKA Exam!

Actual4test.com now offer the **newest CKA exam dumps**, the Actual4test.com CKA exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com CKA dumps with Test Engine here:

https://www.actual4test.com/CKA_examcollection.html (85 Q&As Dumps, **30%OFF** Special

Discount: **Freepdfdumps**)

NEW QUESTION: 32

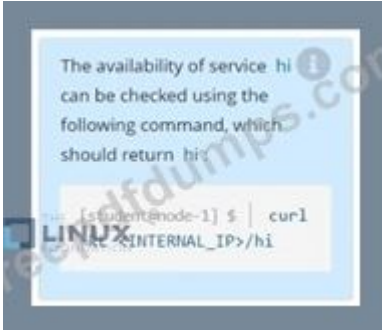
Score: 7%



Task

Create a new nginx Ingress resource as follows:

- * Name: ping
- * Namespace: ing-internal
- * Exposing service hi on path /hi using service port 5678



Answer:

Solution:

```
vi ingress.yaml
```

```
#
```

```
apiVersion: networking.k8s.io/v1
```

```
kind: Ingress
```

```
metadata:
```

```
name: ping
```

```
namespace: ing-internal
```

```
spec:
```

```
rules:
```

```
- http:
```

```
paths:
```

```
- path: /hi
```

```
pathType: Prefix
```

```
backend:
```

```
service:
```

```
name: hi
```

```
port:
```

```
number: 5678
```

```
#
```

```
kubectl create -f ingress.yaml
```

NEW QUESTION: 33

Create a deployment as follows:

- * Name: nginx-random
- * Exposed via a service nginx-random
- * Ensure that the service & pod are accessible via their respective DNS records

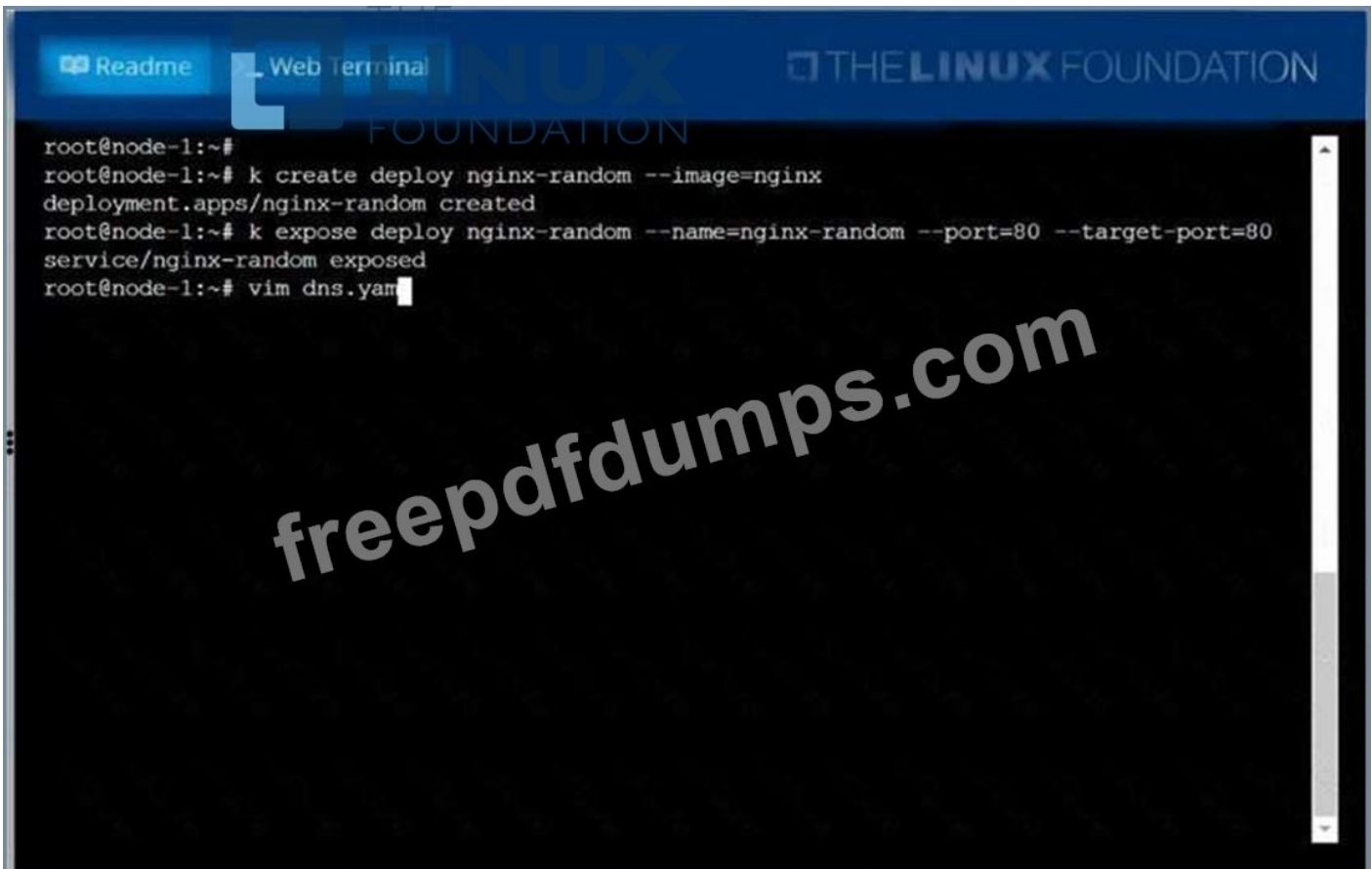
* The container(s) within any pod(s) running as a part of this deployment should use the nginx Image Next, use the utility nslookup to look up the DNS records of the service & pod and write the output to /opt/KUNW00601/service.dns and /opt/KUNW00601/pod.dns respectively.

Answer:

See the solution below.

Explanation

Solution:



The screenshot shows a terminal window with a dark background and light text. At the top, there are navigation links for 'Readme' and 'Web Terminal', and the 'THE LINUX FOUNDATION' logo. The terminal output shows the following commands and their results:

```
root@node-1:~#  
root@node-1:~# k create deploy nginx-random --image=nginx  
deployment.apps/nginx-random created  
root@node-1:~# k expose deploy nginx-random --name=nginx-random --port=80 --target-port=80  
service/nginx-random exposed  
root@node-1:~# vim dns.yaml
```

A large, semi-transparent watermark 'freepdfdumps.com' is overlaid diagonally across the bottom half of the terminal window.

On the NGFW, how can you generate and block a private key from export and thus harden your security posture and prevent rogue administrators or other bad actors from misusing keys?

A. 1) Select Device > Certificate Management > Certificates Device > Certificates

2) Generate the certificate.

3) Select Block Private Key Export.

4) Click Generate to generate the new certificate.

B. 1) Select Device > Certificates

2) Select Certificate Profile

3) Generate the certificate

4) Select Block Private Key Export

C. 1) Select Device > Certificate Management > Certificates > Device > Certificates

2) Import the certificate.

3) Select Import Private Key

4) Click Generate to generate the new certificate.

D. 1) Select Device > Certificates

2) Select Certificate Profile.

3) Generate the certificate

4) Select Block Private Key Export.

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 35

Create an nginx pod with containerPort 80 and it should check the pod running at endpoint / healthz on port 80 and verify and delete the pod.

A. `kubectl run nginx --image=nginx --restart=Always --port=80 --`

`dry-run -o yaml > nginx-pod.yaml`

`// add the livenessProbe section and create`

`apiVersion: v1`

`kind: Pod`

`metadata:`

`labels:`

`containers:`

`- image: nginx`

`name: nginx`

`ports:`

`- containerPort: 60`

`livenessProbe:`

`httpGet:`

`path: /healthz`

`port: 60`

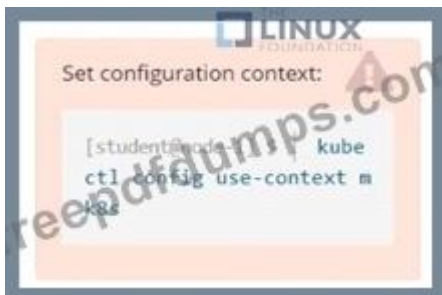
`restartPolicy: Always`

`kubectl create -f nginx-pod.yaml`

```
// verify
kubectl describe pod nginx | grep -i readiness
kubectl delete po nginx
B. kubectl run nginx --image=nginx --restart=Always --port=80 --
dry-run -o yaml > nginx-pod.yaml
// add the livenessProbe section and create
apiVersion: v1
kind: Pod
metadata:
labels:
run: nginx
name: nginx
spec:
containers:
- image: nginx
name: nginx
ports:
- containerPort: 80
livenessProbe:
httpGet:
path: /healthz
port: 80
restartPolicy: Always
kubectl create -f nginx-pod.yaml
// verify
kubectl describe pod nginx | grep -i readiness
kubectl delete po nginx
Answer: B (LEAVE A REPLY)
```

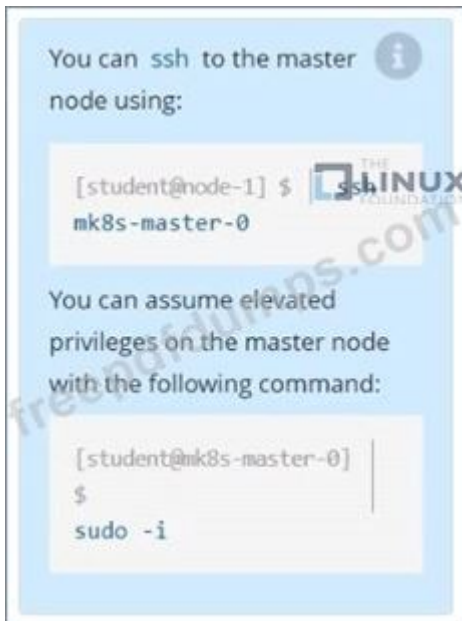
NEW QUESTION: 36

Score: 7%

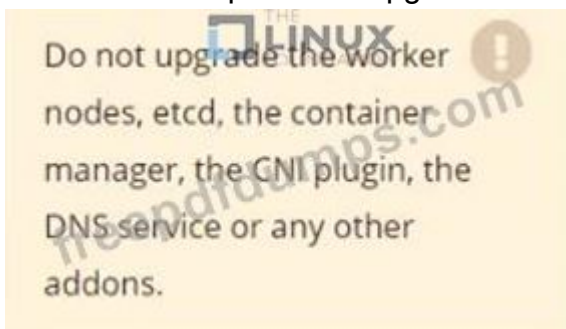


Task

Given an existing Kubernetes cluster running version 1.20.0, upgrade all of the Kubernetes control plane and node components on the master node only to version 1.20.1. Be sure to drain the master node before upgrading it and uncordon it after the upgrade.



You are also expected to upgrade kubelet and kubectl on the master node.



Answer:

SOLUTION:

```
[student@node-1] > ssh ek8s
```

```
kubectl cordon k8s-master
```

```
kubectl drain k8s-master --delete-local-data --ignore-daemonsets --force apt-get install
```

```
kubeadm=1.20.1-00 kubelet=1.20.1-00 kubectl=1.20.1-00 --disableexcludes=kubernetes
```

```
kubeadm upgrade apply 1.20.1 --etcd-upgrade=false systemctl daemon-reload systemctl restart
```

```
kubelet kubectl uncordon k8s-master
```

NEW QUESTION: 37

Get list of PVs and order by size and write to file "/opt/pvstorage.txt"

Answer:

```
kubectl get pv --sort-by=.spec.capacity.storage > /opt/pv storage.txt
```

NEW QUESTION: 38

Schedule a pod as follows:

* Name: nginx-kusc00101

* Image: nginx

* Node selector: disk=ssd

Answer:


```
Readme >... Web Terminal THE LINUX FOUNDATION

root@node-1:~# vim disk.yaml
root@node-1:~# k create -f disk.yaml
pod/nginx-kusc00101 created
root@node-1:~# k get po
NAME                READY   STATUS    RESTARTS   AGE
cpu-utilizer-98b9se 1/1     Running   0           5h59m
cpu-utilizer-ab2d3s 1/1     Running   0           5h59m
cpu-utilizer-kipb9a 1/1     Running   0           5h59m
ds-kusc00201-2r2k9  1/1     Running   0           1m
ds-kusc00201-hzm9q  1/1     Running   0           1m
foo                 1/1     Running   0           6h1m
front-end           1/1     Running   0           6h1m
hungry-bear         1/1     Running   0           9m37s
kucc8               1/1     Running   0           7m37s
nginx-kusc00101     1/1     Running   0           9s
webserver-84c55967f4-qz1w 1/1     Running   0           6h16m
webserver-84c55967f4-t4791 1/1     Running   0           6h16m
root@node-1:~#
```

NEW QUESTION: 39

List all the pods showing name and namespace with a json path expression

Answer:

See the solution below.

Explanation

```
kubectl get pods -o=jsonpath="{.items[*]['metadata.name'], 'metadata.namespace'}"
```

NEW QUESTION: 40

From the pod label name=cpu-utilizer, find pods running high CPU workloads and write the name of the pod consuming most CPU to the file /opt/KUTR00102/KUTR00102.txt (which already exists).

Answer:

solution

kubectl scale deployment webapp -replicas=5 //Verify kubectl get deploy kubectl get po,rs

NEW QUESTION: 42

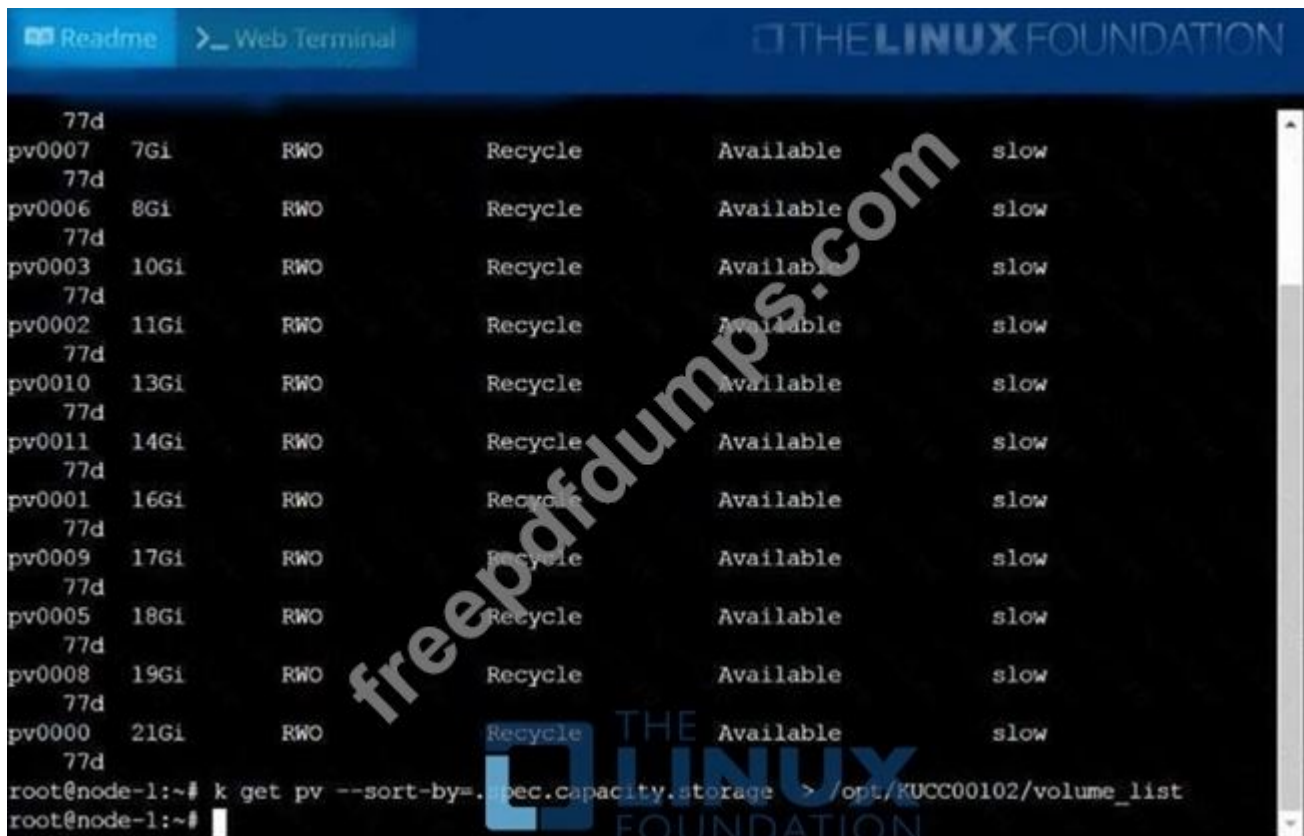
List all persistent volumes sorted by capacity, saving the full kubectl output to /opt/KUCC00102/volume_list. Use kubectl 's own functionality for sorting the output, and do not manipulate it any further.

Answer:

See the solution below.

Explanation

solution



```
77d
pv0007 7Gi      RWO      Recycle   Available slow
77d
pv0006 8Gi      RWO      Recycle   Available slow
77d
pv0003 10Gi     RWO      Recycle   Available slow
77d
pv0002 11Gi     RWO      Recycle   Available slow
77d
pv0010 13Gi     RWO      Recycle   Available slow
77d
pv0011 14Gi     RWO      Recycle   Available slow
77d
pv0001 16Gi     RWO      Recycle   Available slow
77d
pv0009 17Gi     RWO      Recycle   Available slow
77d
pv0005 18Gi     RWO      Recycle   Available slow
77d
pv0008 19Gi     RWO      Recycle   Available slow
77d
pv0000 21Gi     RWO      Recycle   Available slow
77d
root@node-1:~# k get pv --sort-by=.spec.capacity.storage > /opt/KUCC00102/volume_list
root@node-1:~#
```

NEW QUESTION: 43

Create a pod as follows:

- * Name:non-persistent-redis
- * container Image:redis
- * Volume with name:cache-control
- * Mount path:/data/redis

The pod should launch in thestagingnamespace and the volumemust notbe persistent.

Answer:

See the solution below.

Explanation

solution


```
root@node-1:~#  
root@node-1:~#  
root@node-1:~# vim volume.yaml  
root@node-1:~# k create -f volume.yaml  
pod/non-persistent-redis created  
root@node-1:~# k get po -n staging  
NAME                READY   STATUS    RESTARTS   AGE  
non-persistent-redis 1/1     Running   0           6s  
root@node-1:~#
```

The screenshot shows a terminal window with a dark background and light text. At the top, there are navigation tabs for 'Readme' and 'Terminal', and a logo for 'THE LINUX FOUNDATION'. The terminal output shows the user running several Kubernetes commands: 'vim volume.yaml', 'k create -f volume.yaml', and 'k get po -n staging'. The output of the 'k get po' command is a table with columns for NAME, READY, STATUS, RESTARTS, and AGE. The table shows one pod named 'non-persistent-redis' in a 'Running' state with 1/1 ready containers, 0 restarts, and an age of 6s. A large watermark 'freepdfdumps.com' is overlaid diagonally across the terminal output.

NEW QUESTION: 44

Deploy a pod with image=redis on a node with label disktype=ssd

A. // Get list of nodes

```
kubectl get nodes
```

```
//Get node with the label disktype=ssd
```

```
kubectl get no -l disktype=ssd
```

```
// Create a sample yaml file
```

```
kubectl run node-redis --generator=run-pod/v1 --image=redis --dry
```

```
run -o yaml > test-redis.yaml
```

```
// Edit test-redis.yaml file and add nodeSelector
```

```
vim test-redis.yaml
```

```
apiVersion: v1
```

```
- name: node-redis
```

```
image: redis
```

```
imagePullPolicy: IfNotPresent
```

```
kubectl apply -f test-redis.yaml
```

```
// Verify
```

```
K kubectl get po -o wide
```

B. // Get list of nodes

```
kubectl get nodes
```

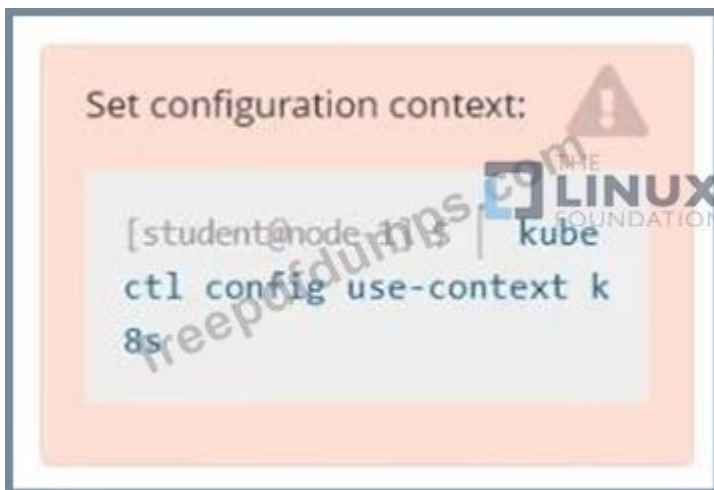
```
//Get node with the label disktype=ssd
```

```
kubectl get no -l disktype=ssd
// Create a sample yaml file
kubectl run node-redis --generator=run-pod/v1 --image=redis --dry
run -o yaml > test-redis.yaml
// Edit test-redis.yaml file and add nodeSelector
vim test-redis.yaml
apiVersion: v1
kind: Pod
metadata:
name: redis
spec:
nodeSelector:
disktype: ssd
containers:
- name: node-redis
image: redis
imagePullPolicy: IfNotPresent
kubectl apply -f test-redis.yaml
// Verify
K kubectl get po -o wide
```

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 45

Score: 4%



Task

Schedule a pod as follows:

- * Name: nginx-kusc00401
- * Image: nginx
- * Node selector: disk=ssd

Answer:

See the solution below.

Explanation

Solution:

```
#yaml
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
name: nginx-kusc00401
```

```
spec:
```

```
containers:
```

```
- name: nginx
```

```
image: nginx
```

```
imagePullPolicy: IfNotPresent
```

```
nodeSelector:
```

```
disk: spinning
```

```
#
```

```
kubectl create -f node-select.yaml
```

NEW QUESTION: 46

Create and configure the service front-end-services so it's accessible through NodePort and routes to the existing pod named front-end.

Answer:

See the solution below.

Explanation

solution

```
Readme >_ Web Terminal THE LINUX FOUNDATION
root@node-1:~# k expose po
error: resource(s) were provided, but no name, label selector, or --all flag specified
See 'kubectl expose -h' for help and examples
root@node-1:~# k expose po front-end --name=front-end-service --port=80 --target-port=80 --t
ype=NodePort
Error from server (NotFound): pods "front-end" not found
root@node-1:~# k expose po front-end --name=front-end-service --port=80 --target-port=80 --
type=NodePort
service/front-end-service exposed
root@node-1:~# k get svc
NAME                TYPE                CLUSTER-IP          EXTERNAL-IP         PORT(S)              AGE
front-end-service   NodePort            10.103.221.227      <none>              80:31828/TCP         3s
kubernetes          ClusterIP           10.96.0.1           <none>              443/TCP              77d
root@node-1:~#
```

Valid CKA Dumps shared by Actual4test.com for Helping Passing CKA Exam!

Actual4test.com now offer the **newest CKA exam dumps**, the Actual4test.com CKA exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com CKA dumps with Test Engine here:

https://www.actual4test.com/CKA_examcollection.html (85 Q&As Dumps, **30%OFF Special Discount: Freepdfdumps**)

NEW QUESTION: 47

Get the memory and CPU usage of all the pods and find out top 3 pods which have the highest usage and put them into the cpusage.txt file

A. // Get the top 3 pods

```
kubectl top pod --all-namespaces | sort --reverse --key 3 --
numeric | head -8
```

// putting into file

```
kubectl top pod --all-namespaces | sort --reverse --key 6 --
numeric | head -6 > cpu-usage.txt
```

// verify

```
cat cpu-usage.txt
```

B. // Get the top 3 pods

```
kubectl top pod --all-namespaces | sort --reverse --key 3 --
numeric | head -3
// putting into file
kubectl top pod --all-namespaces | sort --reverse --key 3 --
numeric | head -3 > cpu-usage.txt
// verify
cat cpu-usage.txt
```

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 48

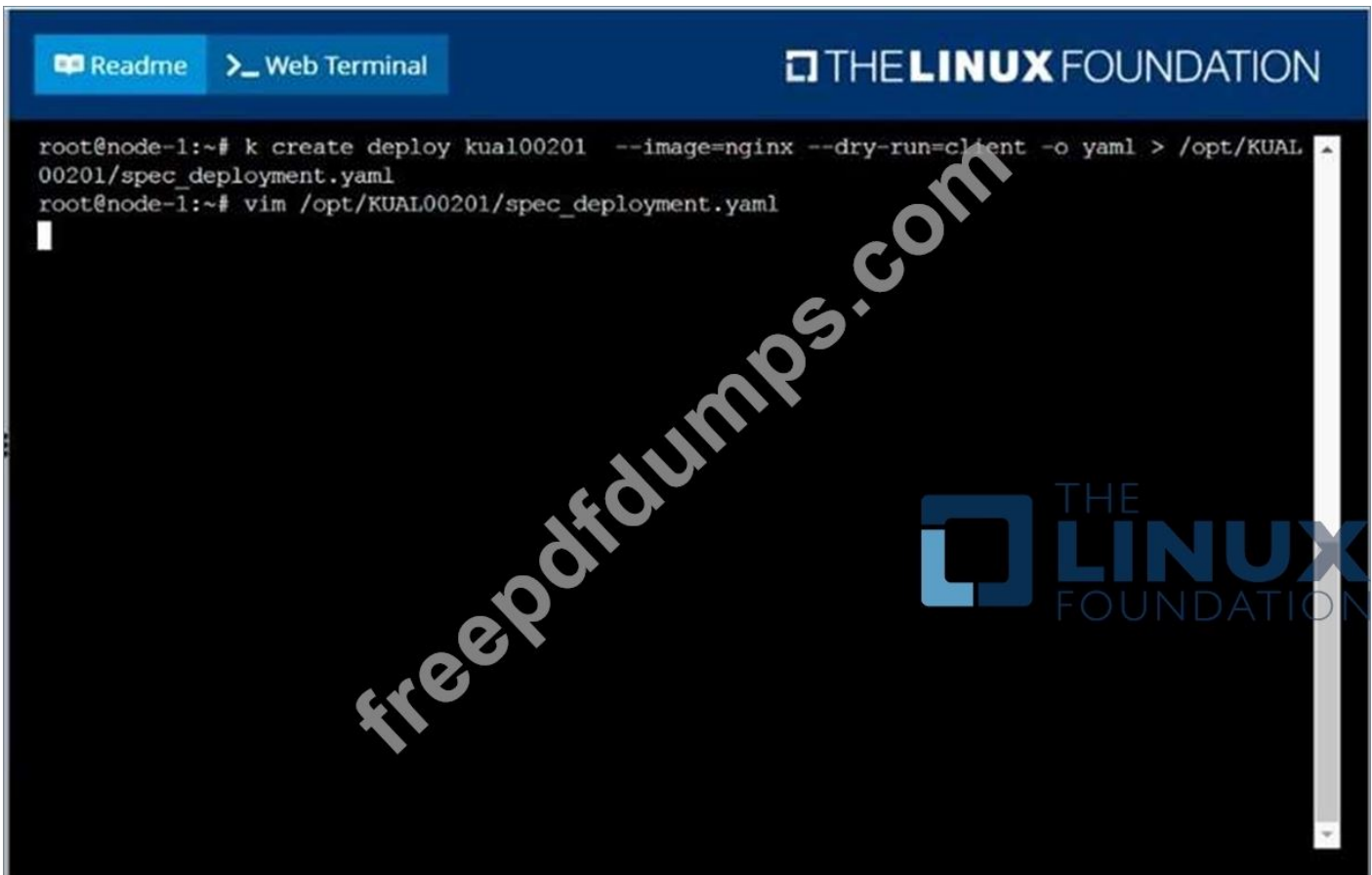
Create a deployment spec file that will:

Launch 7 replicas of the nginx Image with the label `app_runtime_stage=dev` deployment name: `kual00201` Save a copy of this spec file to `/opt/KUAL00201/spec_deployment.yaml` (or `/opt/KUAL00201/spec_deployment.json`).

When you are done, clean up (delete) any new Kubernetes API object that you produced during this task.

Answer:

solution



The screenshot shows a web terminal window with a dark blue header. On the left, there are two tabs: 'Readme' and 'Web Terminal'. On the right, the text 'THE LINUX FOUNDATION' is displayed. The terminal content shows the following commands and output:

```
root@node-1:~# k create deploy kual00201 --image=nginx --dry-run=client -o yaml > /opt/KUAL
00201/spec_deployment.yaml
root@node-1:~# vim /opt/KUAL00201/spec_deployment.yaml
```

A large, diagonal watermark 'freepdfdumps.com' is overlaid across the terminal content. In the bottom right corner, there is a logo for 'THE LINUX FOUNDATION' consisting of a blue square icon and the text 'THE LINUX FOUNDATION'.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app_runtime_stage: dev
  name: kual00201
spec:
  replicas: 7
  selector:
    matchLabels:
      app_runtime_stage: dev
  template:
    metadata:
      labels:
        app_runtime_stage: dev
    spec:
      containers:
      - image: nginx
        name: nginx
~
~
~
~
~
"/opt/KUAL00201/spec_deployment.yaml" 19L, 320C written
```

NEW QUESTION: 49

Create a persistent volume with name app-data, of capacity 2Gi and access mode ReadWriteMany. The type of volume is hostPath and its location is /srv/app-data.

Answer:

See the solution below.

Explanation

solution

Persistent Volume

A persistent volume is a piece of storage in a Kubernetes cluster. PersistentVolumes are a cluster-level resource like nodes, which don't belong to any namespace. It is provisioned by the administrator and has a particular file size. This way, a developer deploying their app on Kubernetes need not know the underlying infrastructure.

When the developer needs a certain amount of persistent storage for their application, the system administrator configures the cluster so that they consume the PersistentVolume provisioned in an easy way.

Creating PersistentVolume

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: app-data
spec:
  capacity: # defines the capacity of PV we are creating
  storage: 2Gi # the amount of storage we are trying to claim
  accessModes: # defines the rights of the volume we are creating
  ReadWriteMany
  hostPath:
    path: "/srv/app-data" # path to which we are creating the volume
```

Challenge

* Create a Persistent Volume named ReadWriteMany, storage classname shared, 2Gi of storage capacity and the host path

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: app-data
spec:
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: /srv/app-data
  storageClassName: shared
```



2. Save the file and create the persistent volume.

Image for post

```
njerry191@cloudshell:~ (extreme-clone-265411) $ kubectl create -f pv.yaml
persistentvolume/pv created
```

3. View the persistent volume.

```
njerry191@cloudshell:~ (extreme-clone-265411) $ kubectl get pv
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS    CLAIM  STORAGECLASS  REASON  AGE
app-data      2Gi       RWX           Retain          Available    
 app-data     2Gi       RWX           Retain          Available  shared      31s
```

* Our persistent volume status is available meaning it is available and it has not been mounted yet. This status will change when we mount the persistentVolume to a persistentVolumeClaim. PersistentVolumeClaim

In a real ecosystem, a system admin will create the PersistentVolume then a developer will create a PersistentVolumeClaim which will be referenced in a pod. A PersistentVolumeClaim is created by specifying the minimum size and the access mode they require from the persistentVolume.

Challenge

* Create a Persistent Volume Claim that requests the Persistent Volume we had created above. The claim should request 2Gi. Ensure that the Persistent Volume Claim has the same storageClassName as the persistentVolume you had previously created.

kind: PersistentVolumeapiVersion: v1metadata:name:

spec:

accessModes:-ReadWriteManyresources:

requests:storage:2Gi

storageClassName:shared

2. Save and create the pvc

```
njerry191@cloudshell:~(extreme-clone-2654111)$ kubectl create -f app-data.yaml
persistentvolumeclaim/app-data created
```

3. View the pvc

Image for post

```
njerry191@cloudshell:~(extreme-clone-2654111)$ kubectl get pvc
NAME      STATUS      VOLUME      CAPACITY      ACCESS MODES      STORAGECLASS
pv        Bound      pv          512m          RWX               shared
```

4. Let's see what has changed in the pv we had initially created.

Image for post

```
njerry191@cloudshell:~(extreme-clone-2654111)$ kubectl get pv
NAME      CAPACITY      ACCESS MODES      RECLAIM POLICY      STATUS      CLAIM      STORAGECLASS      REASON      AGE
pv        512m          RWX               Retain              Bound      default/pv      shared           16m
```

Our status has now changed from available to bound.

5. Create a new pod named myapp with image nginx that will be used to Mount the Persistent Volume Claim with the path /var/app/config.

Mounting a Claim

```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
name: app-data
spec:
  volumes:
  - name: congigpvc
    persistentVolumeClaim:
      claimName: app-data
  containers:
  - image: nginx
    name: app-volume-mounts
    mountPath: "
```

NEW QUESTION: 50

Print all pod name and all image name and write it to a file name "/opt/pod-details.txt"

Answer:

```
kubectl get pods -o=custom-columns='Pod
```

```
Name:metadata.name','Image:spec.containers[*].image' > /opt/pod-details.txt
```

NEW QUESTION: 51

Score: 5%



Task

Monitor the logs of pod bar and:

- * Extract log lines corresponding to error file-not-found
- * Write them to /opt/KUTR00101/bar

Answer:

Solution:

kubectl logs bar | grep 'unable-to-access-website' > /opt/KUTR00101/bar cat /opt/KUTR00101/bar

NEW QUESTION: 52

Create a pod named kucc8 with a single app container for each of the following images running inside (there may be between 1 and 4 images specified):
nginx + redis + memcached.

Answer:

See the solution below.

Explanation

solution



The screenshot shows a terminal window with the following content:

```
root@node-1:~# vim ds.yaml
root@node-1:~# k create -f ds.yaml
daemonset.apps/ds-kusc00201 created
root@node-1:~# k get ds
NAME           DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
ds-kusc00201   2         2         2       2             2           <none>          4s
root@node-1:~# vim /opt/KUCC00108/pod-spec-KUCC00108.yaml
root@node-1:~# k create -f /opt/KUCC00108/pod-spec-KUCC00108.yaml
pod/hungry-bear created
root@node-1:~# k get po
NAME           READY   STATUS    RESTARTS   AGE
cpu-utilizer-98b9se   1/1     Running   0           5h50m
cpu-utilizer-ab2d3s   1/1     Running   0           5h50m
cpu-utilizer-kipb9a   1/1     Running   0           5h50m
ds-kusc00201-2r2k9    1/1     Running   0           4m50s
ds-kusc00201-hzm9q    1/1     Running   0           4m50s
foo              1/1     Running   0           5h52m
front-end        1/1     Running   0           5h52m
hungry-bear       1/1     Running   0           42s
webserver-84c55967f4-qzjcv  1/1     Running   0           6h7m
webserver-84c55967f4-t4791  1/1     Running   0           6h7m
root@node-1:~# k run nginx --image=nginx --dry-run=client -o yaml > nginx.yaml
root@node-1:~# vim nginx.yaml
```


A Kubernetes worker node, named wk8s-node-0 is in state NotReady. Investigate why this is the case, and perform any appropriate steps to bring the node to a Ready state, ensuring that any changes are made permanent.

You can ssh to the failed node using:

```
[student@node-1] $ | ssh Wk8s-node-0
```

You can assume elevated privileges on the node with the following command:

```
[student@w8ks-node-0] $ | sudo -i
```

Answer:

solution



The screenshot shows a web terminal interface with a dark blue header. On the left, there are tabs for 'Readme' and 'Web Terminal'. On the right, the text 'THE LINUX FOUNDATION' is displayed. The terminal content shows the following commands and output:

```
root@node-1:~# kubectl config use-context wk8s
Switched to context "wk8s".
root@node-1:~# k get nodes
NAME           STATUS    ROLES    AGE   VERSION
wk8s-master-0  Ready    master   77d   v1.18.2
wk8s-node-0    NotReady <none>   77d   v1.18.2
wk8s-node-1    Ready    <none>   77d   v1.18.2
root@node-1:~# ssh wk8s-node-0
```

The terminal session then shows a prompt at the root of wk8s-node-0. A large, diagonal watermark 'freepdfdumps.com' is overlaid on the terminal content. At the bottom of the terminal window, the 'THE LINUX FOUNDATION' logo is visible.

```
wk8s-node-0    NotReady    <none>    77d    v1.18.2
wk8s-node-1    Ready      <none>    77d    v1.18.2
root@node-1:~# ssh wk8s-node-0
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1109-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

* Are you ready for Kubernetes 1.19? It's nearly here! Try RC3 with
  sudo snap install microk8s --channel=1.19/candidate --classic
  https://microk8s.io/ has docs and details.
```

```
4 packages can be updated.
1 update is a security update.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

```
student@wk8s-node-0:~$ sudo -i
root@wk8s-node-0:~# systemctl restart kubelet
root@wk8s-node-0:~# systemctl enable kubelet
```

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

```
4 packages can be updated.
1 update is a security update.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

```
student@wk8s-node-0:~$ sudo -i
root@wk8s-node-0:~# systemctl restart kubelet
root@wk8s-node-0:~# systemctl enable kubelet
Created symlink from /etc/systemd/system/multi-user.target.wants/kubelet.service to /lib/systemd/system/kubelet.service.
root@wk8s-node-0:~# ki
logout
student@wk8s-node-0:~$ exit
logout
Connection to 10.250.5.34 closed.
root@node-1:~# k get nodes
NAME           STATUS    ROLES    AGE   VERSION
wk8s-master-0 Ready     master   77d   v1.18.2
wk8s-node-0    Ready     <none>   77d   v1.18.2
wk8s-node-1    Ready     <none>   77d   v1.18.2
root@node-1:~#
```

Create a pod with init container which waits for a service called "myservice" to be created. Once init container completes, the myapp-container should start and print a message "The app is running" and sleep for 3600 seconds.

A. vim multi-container-pod.yaml

```
apiVersion: v1
kind: Pod
metadata:
name: myapp-pod
labels:
app: myapp
spec:
containers:
- name: myapp-container
image: busybox:1.28
command: ['sh', '-c', 'echo The app is running! && sleep
3600']
initContainers:
- name: init-myservice
done"]
// Check whether service called "myservice" exists
```

```
kubectl get svc
```

Note: Pod will not start if service called "myservice" doesn't exist.

```
// Now, Create the pod
```

```
kubectl apply -f multi-container-pod.yaml
```

B. vim multi-container-pod.yaml

```
apiVersion: v1
kind: Pod
metadata:
name: myapp-pod
labels:
app: myapp
spec:
containers:
- name: myapp-container
image: busybox:1.28
command: ['sh', '-c', 'echo The app is running! && sleep
3600']
initContainers:
- name: init-myservice
image: busybox:1.28
```

```
command: ['sh', '-c', "until nslookup myservice.$(cat
/var/run/secrets/kubernetes.io/serviceaccount/namespace).s
vc.cluster.local; do echo waiting for myservice; sleep 2;
done"]
```

```
// Check whether service called "myservice" exists
```

```
kubectl get svc
```

Note: Pod will not start if service called "myservice" doesn't exist.

```
// Now, Create the pod
```

```
kubectl apply -f multi-container-pod.yaml
```

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 55

Install a kubernetes cluster with one master and one worker using kubeadm

A. This is a straightforward question, you need to install kubernetes cluster using kubeadm with one master and one worker.

Installation is considered success once both master and worker nodes become available.

Refer : <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/>

B. This is a straightforward question, you need to install kubernetes cluster using kubeadm with one master and one worker.

Refer : <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/>

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 56

Create a nginx pod that will be deployed to node with the label "gpu=true"

A. `kubectl run nginx --image=nginx --restart=Always --dry-run -o yml > nodeselector-pod.yaml`

```
// add the nodeSelector like below and create the pod
```

```
kubectl apply -f nodeselector-pod.yaml
```

```
vim nodeselector-pod.yaml
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
name: nginx
```

```
spec:
```

```
nodeSelector:
```

```
gpu: true
```

```
containers:
```

```
- image: nginx
```

```
name: nginx
restartPolicy: Always
kubectl apply -f nodeselector-pod.yaml
//Verify
kubectl get no -show-labels
kubectl get po
kubectl describe po nginx | grep Node-Selectors
B. kubectl run nginx --image=nginx --restart=Always --dry-run -o
yaml > nodeselector-pod.yaml
// add the nodeSelector like below and create the pod
kubectl apply -f nodeselector-pod.yaml
vim nodeselector-pod.yaml
apiVersion: v1
kind: Pod
metadata:
name: nginx
spec:
nodeSelector:
gpu: true
yaml
//Verify
kubectl get no -show-labels
kubectl get po
kubectl describe po nginx | grep Node-Selectors
```

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 57

Score:7%



Context

An existing Pod needs to be integrated into the Kubernetes built-in logging architecture (e. g. kubectl logs).

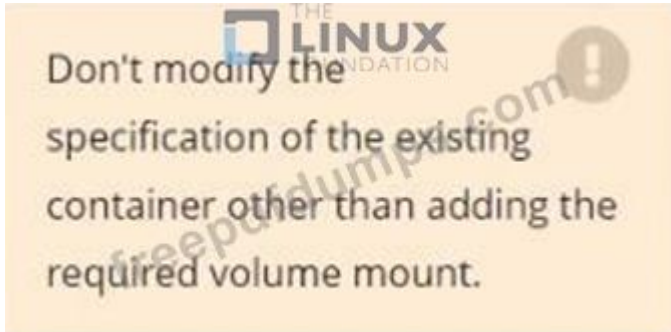
Adding a streaming sidecar container is a good and common way to accomplish this requirement.

Task

Add a sidecar container named sidecar, using the busybox Image, to the existing Pod big-corp-app. The new sidecar container has to run the following command:

```
/bin/sh -c tail -n+1 -f /var/log/big-corp-app.log
```

Use a Volume, mounted at /var/log, to make the log file big-corp-app.log available to the sidecar container.



Answer:

See the solution below.

Explanation

Solution:

```
#
kubectl get pod big-corp-app -o yaml
#
apiVersion: v1
kind: Pod
metadata:
name: big-corp-app
spec:
containers:
- name: big-corp-app
image: busybox
args:
- /bin/sh
- -c
- >
i=0;
while true;
do
echo "$(date) INFO $i" >> /var/log/big-corp-app.log;
i=$((i+1));
sleep 1;
done
volumeMounts:
- name: logs
```

```
mountPath: /var/log
- name: count-log-1
image: busybox
args: [/bin/sh, -c, 'tail -n+1 -f /var/log/big-corp-app.log']
volumeMounts:
- name: logs
mountPath: /var/log
volumes:
- name: logs
emptyDir: {
}
#
kubectl logs big-corp-app -c count-log-1
```

NEW QUESTION: 58

Create a pod with init container which create a file "test.txt" in "workdir" directory. Main container should check a file "test.txt" exists and execute sleep 9999 if the file exists.

A. // create an initial yaml file with this

```
kubectl run init-cont-pod --image=alpine --restart=Never --dry-run -o
```

```
yaml > init-cont-pod.yaml
```

```
// edit the yml as below and create it
```

```
vim init-cont-pod.yaml
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
name: init-cont-pod
```

```
labels:
```

```
app: myapp
```

```
spec:
```

```
volumes:
```

```
- name: test-volume
```

```
emptyDir: {}
```

```
containers:
```

```
- name: main-container
```

```
image: busybox:1.28
```

```
command: ['sh', '-c', 'if [ -f /workdir/test.txt ]; then sleep 9999; fi']
```

```
volumeMounts:
```

```
- name: test-volume
```

```
mountPath: /workdir
```

initContainers:

- name: init-myservice

image: busybox:1.28

command: ['sh', '-c', "mkdir /workdir; echo > /workdir/test.txt"]

volumeMounts:

- name: test-volume

mountPath: /workdir

// Create the pod

kubectl apply -f init-cont-pod.yaml

kubectl get pods

// Check Events by doing

kubectl describe po init-cont-pod

Init Containers:

init-myservice:

Container ID:

docker://ebdbf5fad1c95111d9b0e0e2e743c2e347c81b8d4eb5abcccdfe1dd745240d4f

Image: busybox:1.28

Image ID: dockerpullable://busybox@sha256:141c253bc4c3fd0a201d32dc1f493bcf3fff003b6df416dea4f41046e0f37d47

Port: <none>

Host Port: <none>

Command:

sh

-c

mkdir /workdir; echo > /workdir/test.txt

State: Terminated Reason: Completed

B. // create an initial yaml file with this

kubectl run init-cont-pod --image=alpine --restart=Never --dry-run -o

yaml > init-cont-pod.yaml

// edit the yml as below and create it

vim init-cont-pod.yaml

apiVersion: v1

kind: Pod

metadata:

name: init-cont-pod

labels:

app: myapp

spec:

volumes:

```
- name: test-volume
emptyDir: {}
containers:
- name: main-container
image: busybox:1.28
command: ['sh', '-c', 'if [ -f /workdir/test.txt ]; then sleep
9999; fi']
volumeMounts:
image: busybox:1.28
command: ['sh', '-c', "mkdir /workdir; echo >
/workdir/test.txt"]
volumeMounts:
- name: test-volume
mountPath: /workdir
// Create the pod
kubectl apply -f init-cont-pod.yaml
kubectl get pods
// Check Events by doing
kubectl describe po init-cont-pod
Init Containers:
init-myservice:
Container ID:
docker://ebdbf5fad1c95111d9b0e0e2e743c2e347c81b8d4eb5abcccdfe1dd74524
0d4f
Image: busybox:1.28
Image ID: dockerpullable://busybox@sha256:141c253bc4c3fd0a201d32dc1f493bcf3fff003b6df
416dea4f41046e0f37d47
Port: <none>
Host Port: <none>
Command:
sh
-c
mkdir /workdir; echo > /workdir/test.txt
State: Terminated Reason: Completed
```

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 59

Score: 7%



Task

Create a new NetworkPolicy named allow-port-from-namespace in the existing namespace echo. Ensure that the new NetworkPolicy allows Pods in namespace my-app to connect to port 9000 of Pods in namespace echo.

Further ensure that the new NetworkPolicy:

- * does not allow access to Pods, which don't listen on port 9000
- * does not allow access from Pods, which are not in namespace my-app

Answer:

See the solution below.

Explanation

Solution:

```
#network.yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-port-from-namespace
  namespace: internal
spec:
  podSelector:
    matchLabels: {
    }
  policyTypes:
  - Ingress
  ingress:
  - from:
  - podSelector: {
  }
  ports:
  - protocol: TCP
    port: 8080
#spec.podSelector namespace pod
kubectl create -f network.yaml
```

NEW QUESTION: 60

Create the service as type NodePort with the port 32767 for the nginx pod with the pod selector app: my-nginx

Answer:

```
kubectl run nginx --image=nginx --restart=Never -- labels=app=nginx --port=80 --dry-run -o yaml > nginx-pod.yaml
```

NEW QUESTION: 61

Create a snapshot of the etcd instance running at https://127.0.0.1:2379, saving the snapshot to the file path

/srv/data/etcd-snapshot.db.

The following TLS certificates/key are supplied for connecting to the server with etcdctl:

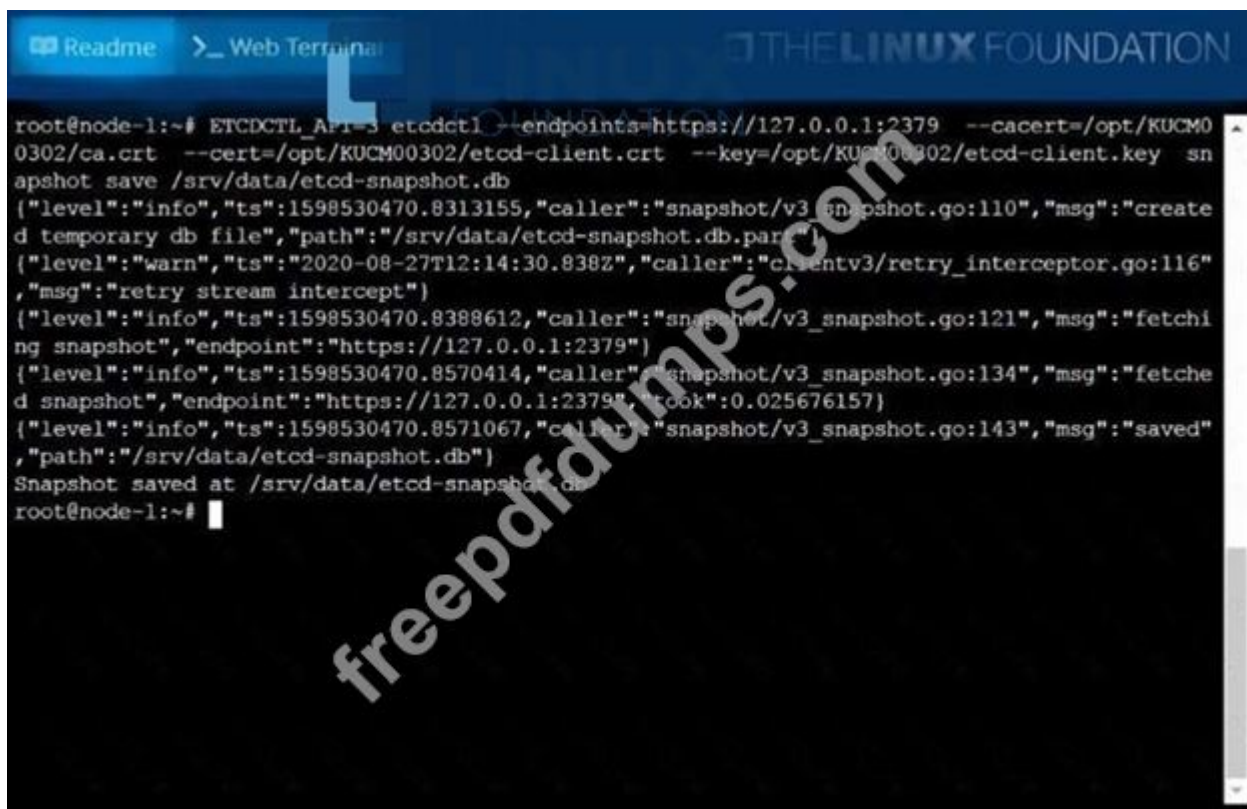
- * CA certificate: /opt/KUCM00302/ca.crt
- * Client certificate: /opt/KUCM00302/etcd-client.crt
- * Client key: /opt/KUCM00302/etcd-client.key

Answer:

See the solution below.

Explanation

solution



```
root@node-1:~# ETCDCTL_API=3 etcdctl --endpoints=https://127.0.0.1:2379 --cacert=/opt/KUCM00302/ca.crt --cert=/opt/KUCM00302/etcd-client.crt --key=/opt/KUCM00302/etcd-client.key snapshot save /srv/data/etcd-snapshot.db
{"level":"info","ts":1598530470.8313155,"caller":"snapshot/v3_snapshot.go:110","msg":"created temporary db file","path":"/srv/data/etcd-snapshot.db.part"}
{"level":"warn","ts":"2020-08-27T12:14:30.838Z","caller":"clientv3/retry_interceptor.go:116","msg":"retry stream intercept"}
{"level":"info","ts":1598530470.8388612,"caller":"snapshot/v3_snapshot.go:121","msg":"fetching snapshot","endpoint":"https://127.0.0.1:2379"}
{"level":"info","ts":1598530470.8570414,"caller":"snapshot/v3_snapshot.go:134","msg":"fetched snapshot","endpoint":"https://127.0.0.1:2379","took":0.025676157}
{"level":"info","ts":1598530470.8571067,"caller":"snapshot/v3_snapshot.go:143","msg":"saved","path":"/srv/data/etcd-snapshot.db"}
Snapshot saved at /srv/data/etcd-snapshot.db
root@node-1:~#
```

Valid CKA Dumps shared by Actual4test.com for Helping Passing CKA Exam!

Actual4test.com now offer the **newest CKA exam dumps**, the Actual4test.com CKA exam questions have been updated and answers have been corrected get the **newest**

Actual4test.com CKA dumps with Test Engine here:

https://www.actual4test.com/CKA_examcollection.html (85 Q&As Dumps, **30%OFF Special**

Discount: Freepdfdumps)

NEW QUESTION: 62

Score: 4%



Task

Scale the deployment presentation

Answer:

See the solution below.

Explanation

Solution:

```
kubectl get deployment
```

```
kubectl scale deployment.apps/presentation --replicas=6
```

NEW QUESTION: 63

Create a PersistentVolumeClaim of at least 3Gi storage and access mode ReadWriteOnce and verify status is Bound

A. vim task-pv-claim.yaml

```
apiVersion: v1
```

```
kind: PersistentVolumeClaim
```

```
metadata:
```

```
name: task-pv-claim
```

```
spec:
```

```
storageClassName: ""
```

```
accessModes:
```

```
- ReadWriteOnce
```

```
resources:
```

```
requests:
```

```
storage: 3Gi
```

```
kubectl apply -f task-pv-claim.yaml
```

```
//Verify
```

```
kubectl get pv
```

```
NAME CAPACITY ACCESS
MODES RECLAIM POLICY STATUS CLAIM
STORAGECLASS REASON AGE
task-pv-volume 5Gi RWO
Retain Bound default/task-pv-claim
6m16s
kubectl get pvc
NAME STATUS VOLUME
CAPACITY ACCESS MODES STORAGECLASS AGE
task-pv-claim Bound task-pv-volume
5Gi RWO 6s
```

B. vim task-pv-claim.yaml

```
apiVersion: v2
```

```
kind: PersistentVolumeClaim
```

```
metadata:
```

```
name: task-pv-claim
```

```
spec:
```

```
storageClassName: ""
```

```
accessModes:
```

```
- ReadWriteOnce
```

```
resources:
```

```
requests:
```

```
storage: 4Gi
```

```
kubectl apply -f task-pv-claim.yaml
```

```
//Verify
```

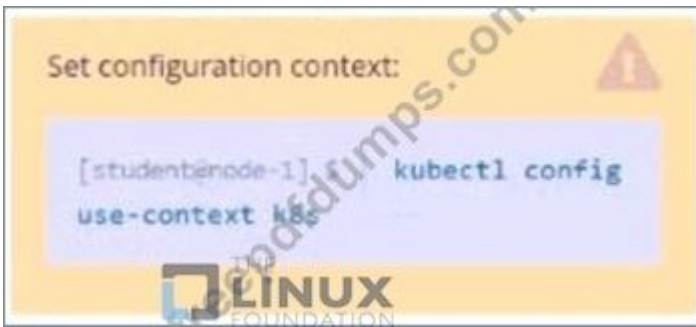
```
kubectl get pv
```

```
NAME CAPACITY ACCESS
MODES RECLAIM POLICY STATUS CLAIM
STORAGECLASS REASON AGE
task-pv-volume 4Gi RWO
Retain Bound default/task-pv-claim
6m16s
kubectl get pvc
NAME STATUS VOLUME
CAPACITY ACCESS MODES STORAGECLASS AGE
task-pv-claim Bound task-pv-volume
5Gi RWO 6s
```

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 64

Task Weight: 4%



Task

Scale the deployment webserver to 3 pods.

Answer:

Solution:

```
student@node-1:~$ kubectl scale deploy webserver --replicas=3
deployment.apps/webserver scaled
student@node-1:~$ kubectl scale deploy webserver --replicas=3
```

NEW QUESTION: 65

Ensure a single instance of pod nginx is running on each node of the Kubernetes cluster where nginx also represents the Image name which has to be used. Do not override any taints currently in place.

Use DaemonSet to complete this task and use ds-kusc00201 as DaemonSet name.

Answer:

solution



```

apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: fluentd-elasticsearch
  namespace: kube-system
  labels:
    k8s-app: fluentd-logging
spec:
  selector:
    matchLabels:
      name: fluentd-elasticsearch
  template:
    metadata:
      labels:
        name: fluentd-elasticsearch
    spec:
      tolerations:
        # this toleration is to make the daemonset runnable on master nodes
        # remove it if your masters can't run pods
        - key: node-role.kubernetes.io/master
          effect: NoSchedule
      containers:
        - name: nginx
          image: nginx
-- INSERT --

```

17,19

All

```

apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: ds-kusc00201
spec:
  selector:
    matchLabels:
      name: fluentd-elasticsearch
  template:
    metadata:
      labels:
        name: fluentd-elasticsearch
    spec:
      containers:
        - name: nginx
          image: nginx

```

:wq



```
Readme Web Terminal THE LINUX FOUNDATION
root@node-1:~# vim ds.yaml
root@node-1:~# k create -f ds.yaml
daemonset.apps/ds-kusc00201 created
root@node-1:~# k get ds
NAME          DESIRED  CURRENT  READY  UP-TO-DATE  AVAILABLE  NODE SELECTOR  AGE
ds-kusc00201  2        2        2      2           2          <none>         4s
root@node-1:~#
```

NEW QUESTION: 66

Perform the following tasks:

Add an init container to hungry-bear (which has been defined in spec file /opt/KUCC00108/pod-spec-KUC C00108.yaml) The init container should create an empty file named /workdir/calm.txt If /workdir/calm.txt is not detected, the pod should exit Once the spec file has been updated with the init container definition, the pod should be created

Answer:

solution

```
root@node-1:~# vim ds.yaml
iroot@node-1:~# k create -f ds.yaml
daemonset.apps/ds-kusc00201 created
root@node-1:~# k get ds
NAME                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
ds-kusc00201        2         2         2       2             2           <none>          4s
root@node-1:~# vim /opt/KUCC00108/pod-spec-KUCC00108.yaml
```



```
apiVersion: v1
kind: Pod
metadata:
  name: hungry-bear
spec:
  volumes:
  - name: workdir
    emptyDir: {}
  containers:
  - name: checker
    image: alpine
    command: ["/bin/sh", "-c", "if [ ! -f /workdir/calm.txt ];
              then sleep 100000; else exit 1; fi"]
    volumeMounts:
    - name: workdir
      mountPath: /workdir
  initContainers:
  - name: create
    image: alpine
    command: ["/bin/sh", "-c", "touch /workdir/calm.txt"]
    volumeMounts:
    - name: workdir
      mountPath: /workdir
```

```
:wq
```

```
root@node-1:~# vim ds.yaml
iroot@node-1:~# k create -f ds.yaml
daemonset.apps/ds-kusc00201 created
root@node-1:~# k get ds
NAME                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
ds-kusc00201        2         2         2       2            2           <none>          4s
root@node-1:~# vim /opt/KUCC00108/pod-spec-KUCC00108.yaml
root@node-1:~# k create -f /opt/KUCC00108/pod-spec-KUCC00108.yaml
pod/hungry-bear created
root@node-1:~#
```

NEW QUESTION: 67

Create a busybox pod that runs the command "env" and save the output to "envpod" file

Answer:

See the solution below.

Explanation

```
kubectl run busybox --image=busybox --restart=Never --rm -it -- env > envpod.yaml
```

NEW QUESTION: 68

Ensure a single instance of pod nginx is running on each node of the Kubernetes cluster where nginx also represents the Image name which has to be used. Do not override any taints currently in place.

Use DaemonSet to complete this task and use ds-kusc00201 as DaemonSet name.

Answer:

See the solution below.

Explanation

solution

```
root@node-1:~# vim ds.yaml
i
```



```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: fluentd-elasticsearch
  namespace: kube-system
  labels:
    k8s-app: fluentd-logging
spec:
  selector:
    matchLabels:
      name: fluentd-elasticsearch
  template:
    metadata:
      labels:
        name: fluentd-elasticsearch
    spec:
      tolerations:
        # this toleration is to have the daemonset runnable on master nodes
        # remove it if your masters can't run pods
        - key: node-role.kubernetes.io/master
          effect: NoSchedule
      containers:
        - name: nginx
          image: nginx
-- INSERT --
```

```
Readme Web Terminal THE LINUX FOUNDATION
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: ds-kusc00201
spec:
  selector:
    matchLabels:
      name: fluentd-elasticsearch
  template:
    metadata:
      labels:
        name: fluentd-elasticsearch
    spec:
      containers:
      - name: nginx
        image: nginx
```

```
Readme Web Terminal THE LINUX FOUNDATION
root@node-1:~# vim ds.yaml
iroot@node-1:~# k create -f ds.yaml
daemonset.apps/ds-kusc00201 created
root@node-1:~# k get ds
NAME           DESIRED  CURRENT  READY  UP-TO-DATE  AVAILABLE  NODE SELECTOR  AGE
ds-kusc00201   2        2        2      2            2           <none>         4s
root@node-1:~#
```

NEW QUESTION: 69

What is the maximum number of samples that can be submitted to WildFire manually per day?

- A. 2,000
- B. 15,000

C. 1,000

D. 5,000

Answer: ([SHOW ANSWER](#))

Valid CKA Dumps shared by Actual4test.com for Helping Passing CKA Exam!

Actual4test.com now offer the **newest CKA exam dumps**, the Actual4test.com CKA exam **questions have been updated** and **answers have been corrected** get the **newest**

Actual4test.com CKA dumps with Test Engine here:

https://www.actual4test.com/CKA_examcollection.html (**85** Q&As Dumps, **30%OFF** Special

Discount: **Freepdfdumps**)