



CKA^{Q&As}

Certified Kubernetes Administrator (CKA) Program

Pass Linux Foundation CKA Exam with 100% Guarantee

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

<https://www.passapply.com/cka.html>

100% Passing Guarantee
100% Money Back Assurance

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

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers





QUESTION 1

Create a nginx pod with label env=test in engineering namespace .

Correct Answer: Check the answer in explanation.

```
kubectl run nginx --image=nginx --restart=Never --labels=env=test --namespace=engineering --dry-run -o yaml > nginx-pod.yaml  
kubectl run nginx --image=nginx --restart=Never --labels=env=test --namespace=engineering --dry-run -o yaml | kubectl create -n engineering -f YAML File:
```

```
apiVersion: v1 kind: Pod metadata: name: nginx namespace: engineering labels: env: test spec: containers:
```

```
-name: nginx image: nginx imagePullPolicy: IfNotPresent restartPolicy: Never
```

```
kubectl create -f nginx-pod.yaml
```

QUESTION 2

CORRECT TEXT

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

Don't modify the specification of the existing container other than adding the required volume mount.

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 .

Correct Answer:

```
#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
```



QUESTION 3

Create 2 nginx image pods in which one of them is labelled with env=prod and another one labelled with env=dev and verify the same.

Correct Answer: Check the answer in explanation.

Solution

kubectl run --generator=run-pod/v1 --image=nginx -- labels=env=prod nginx-prod --dry-run -o yaml > nginx-prodpod.yaml Now, edit nginx-prod-pod.yaml file and remove entries like "creationTimestamp":

```
null" "dnsPolicy: ClusterFirst"
```

```
vim nginx-prod-pod.yaml
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
labels:
```

```
env: prod
```

```
name: nginx-prod
```

```
spec:
```

```
containers:
```

```
-
```

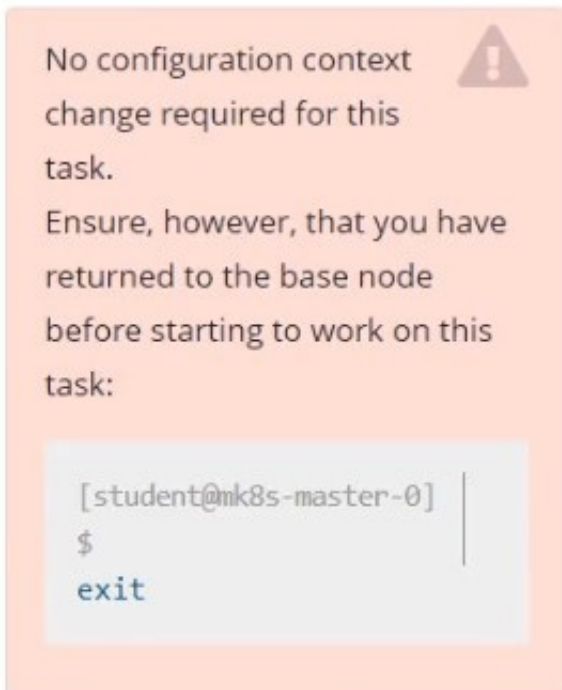
```
image: nginx name: nginx-prod restartPolicy: Always # kubectl create -f nginx-prod-pod.yaml kubectl run --generator=run-pod/v1 --image=nginx -labels=env=dev nginx-dev --dry-run -o yaml > nginx-dev-pod.yaml apiVersion: v1 kind: Pod metadata: labels: env: dev name: nginx-dev spec: containers:
```

```
-
```

```
image: nginx name: nginx-dev restartPolicy: Always # kubectl create -f nginx-prod-dev.yaml Verify : kubectl get po --show-labels kubectl get po -l env=prod kubectl get po -l env=dev
```

QUESTION 4

SIMULATION



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`:

1.

CA certificate: `/opt/KUCM00302/ca.crt`

2.

Client certificate: `/opt/KUCM00302/etcd-client.crt`

3.

Client key: `/opt/KUCM00302/etcd-client.key`

Correct Answer: Check the answer in explanation.

Solution



```
Readme >_ Web Terminal THE LINUX FOUNDATION

root@node-1:~# ETCDCCTL_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:~#
```

QUESTION 5

SIMULATION

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

Correct Answer: Check the answer in 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 fron-end --name=front-end-service --port=80 --target-port=80 --t
ype=NodePort
Error from server (NotFound): pods "fron-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:~#
```

[CKA VCE Dumps](#)

[CKA Exam Questions](#)

[CKA Braindumps](#)