

# Certified Kubernetes Administrator (CKA) Program Exam

## Linux Foundation CKA Dumps Available Here at:

<https://www.certification-questions.com/linux-foundation-exam/cka-dumps.html>

Enrolling now you will get access to 53 questions in a unique set of CKA dumps

### Question 1

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.

#### Options:

A. Option is correct.  
See the solution below.

#### Explanation:

```
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
```

**Answer: A**

**Explanation:**

rrrect.

See the solution below.

Explanation:

```
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 2

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

**Options:**

A. Option is correct.

See the solution below.

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 -nengineering-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

**Answer: A**

**Explanation:**

rrrect.

See the solution below.

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 -nengineering-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 3

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

#### Options:

A. Option is correct.  
See the solution below.

Explanation:

```
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: A**

#### Explanation:

rrrect.  
See the solution below.

Explanation:

```
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
```

## Question 4

List all the pods sorted by name

### Options:

A. Option is correct.  
See the solution below.

Explanation:

```
kubect1 get pods --sort-by=.metadata.name
```

**Answer: A**

### Explanation:

rract.  
See the solution below.

Explanation:

```
kubect1 get pods --sort-by=.metadata.name
```

## Question 5

Create a pod with image nginx called nginx and allow traffic on port 80

### Options:

A. Option is correct.  
See the solution below.

Explanation:

```
kubectlrn nginx --image=nginx --restart=Never --port=80
```

**Answer: A**

**Explanation:**

rect.

See the solution below.

Explanation:

kubectlrn nginx --image=nginx --restart=Never --port=80

## Question 6

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

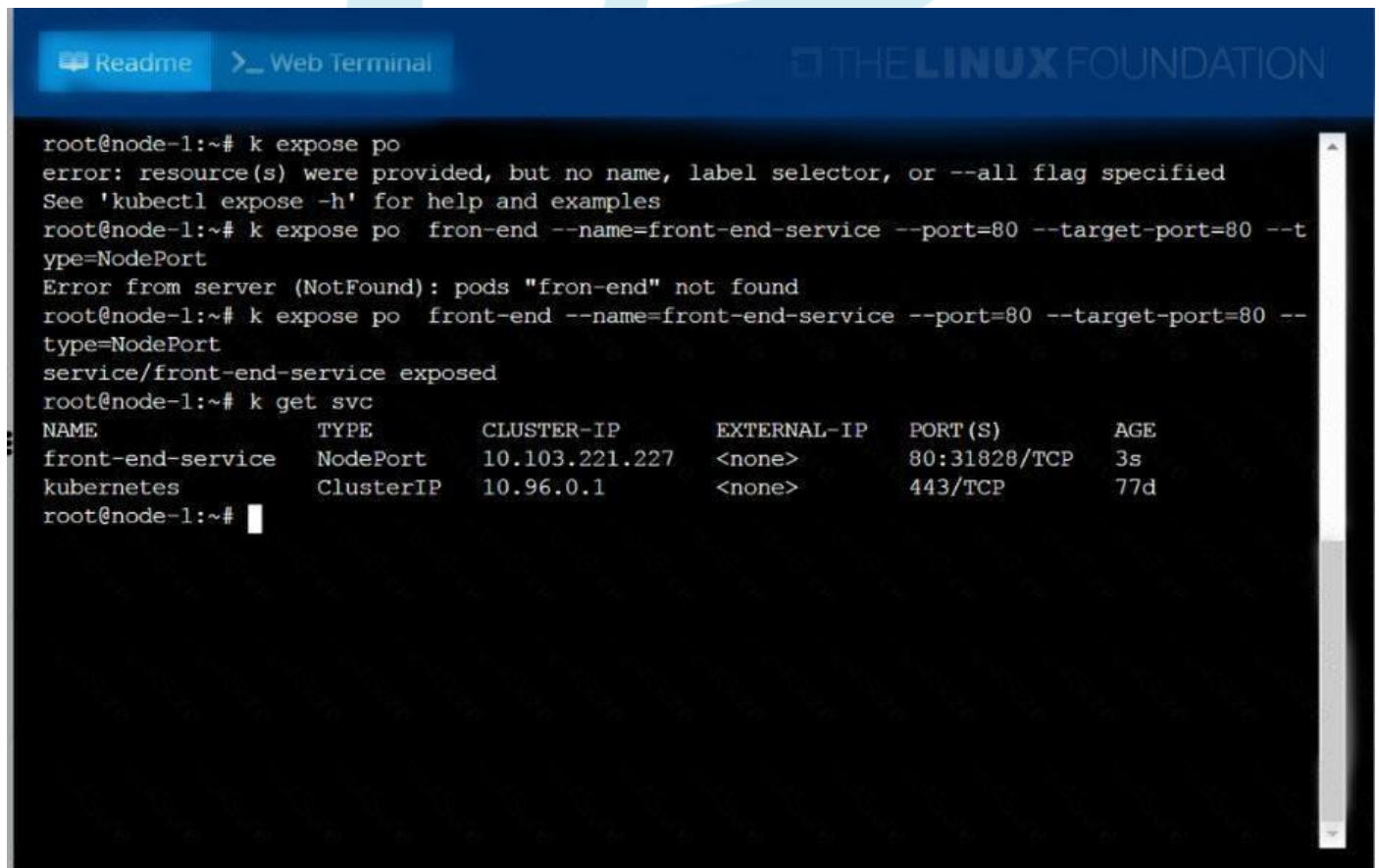
**Options:**

A. Option 1 is correct.

See the solution below.

Explanation:

solution



```
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:~#
```

**Answer: A**

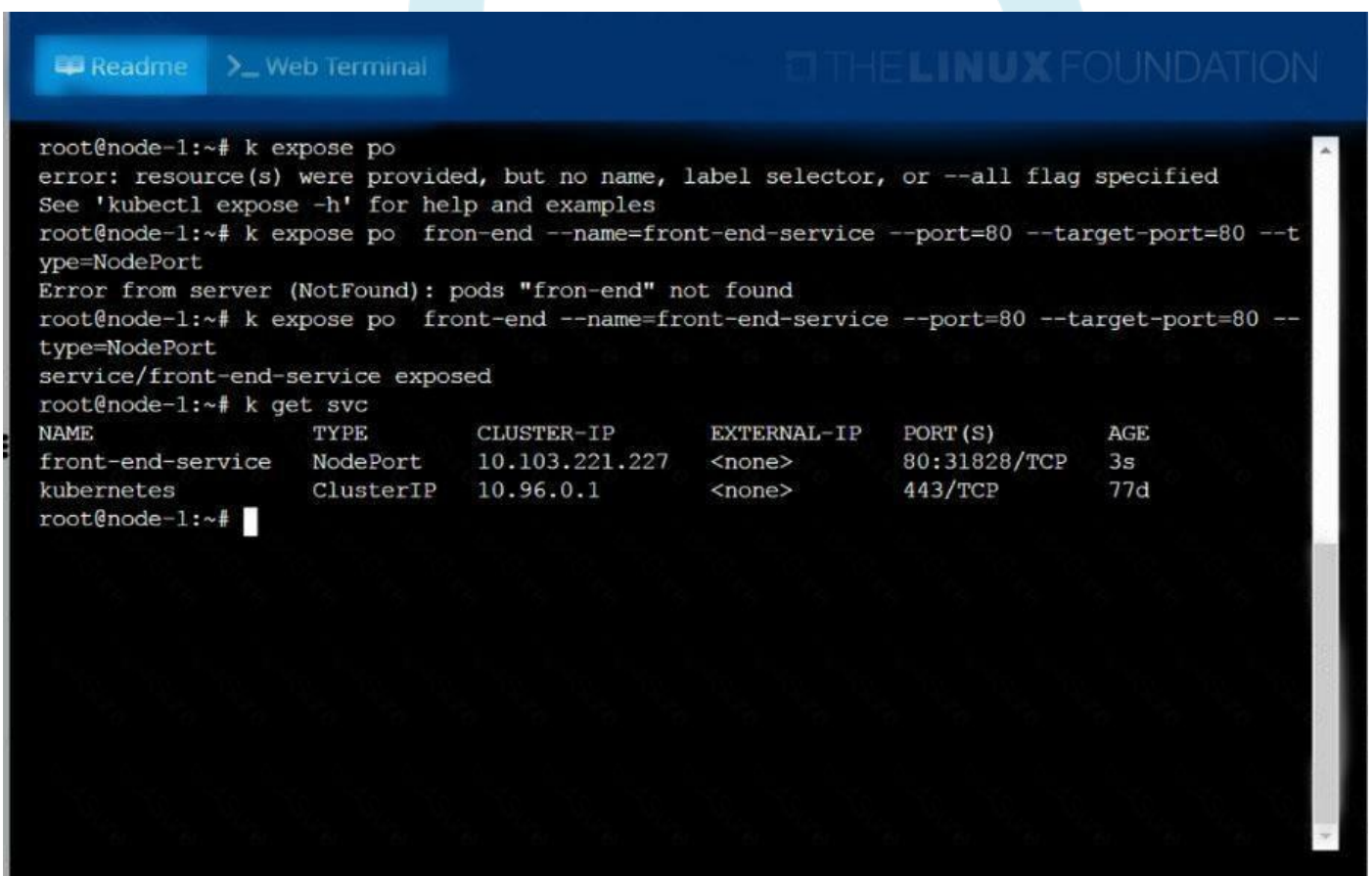
**Explanation:**

rrrect.

See the solution below.

Explanation:

solution



```
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:~#
```

## Question 7

Set the node named `dek8s-node-1` as unavailable and reschedule all the pods running on it.

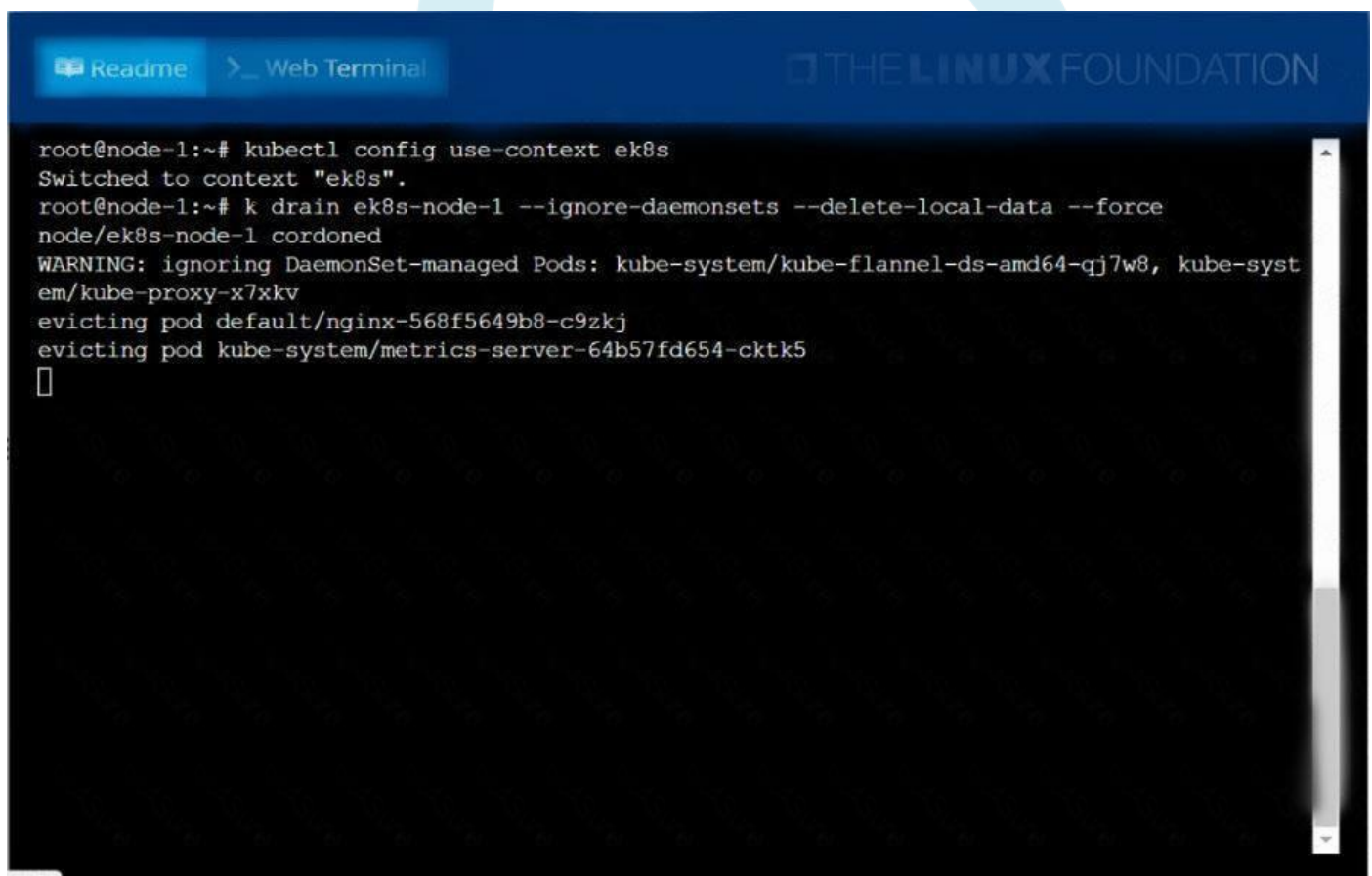
**Options:**



A. Option is correct.  
See the solution below.

Explanation:

solution



The screenshot shows a terminal window with a dark background and light blue text. The terminal is titled 'Web Terminal' and has 'THE LINUX FOUNDATION' logo in the top right corner. The user is logged in as 'root' on 'node-1'. The commands and output are as follows:

```
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
█
```

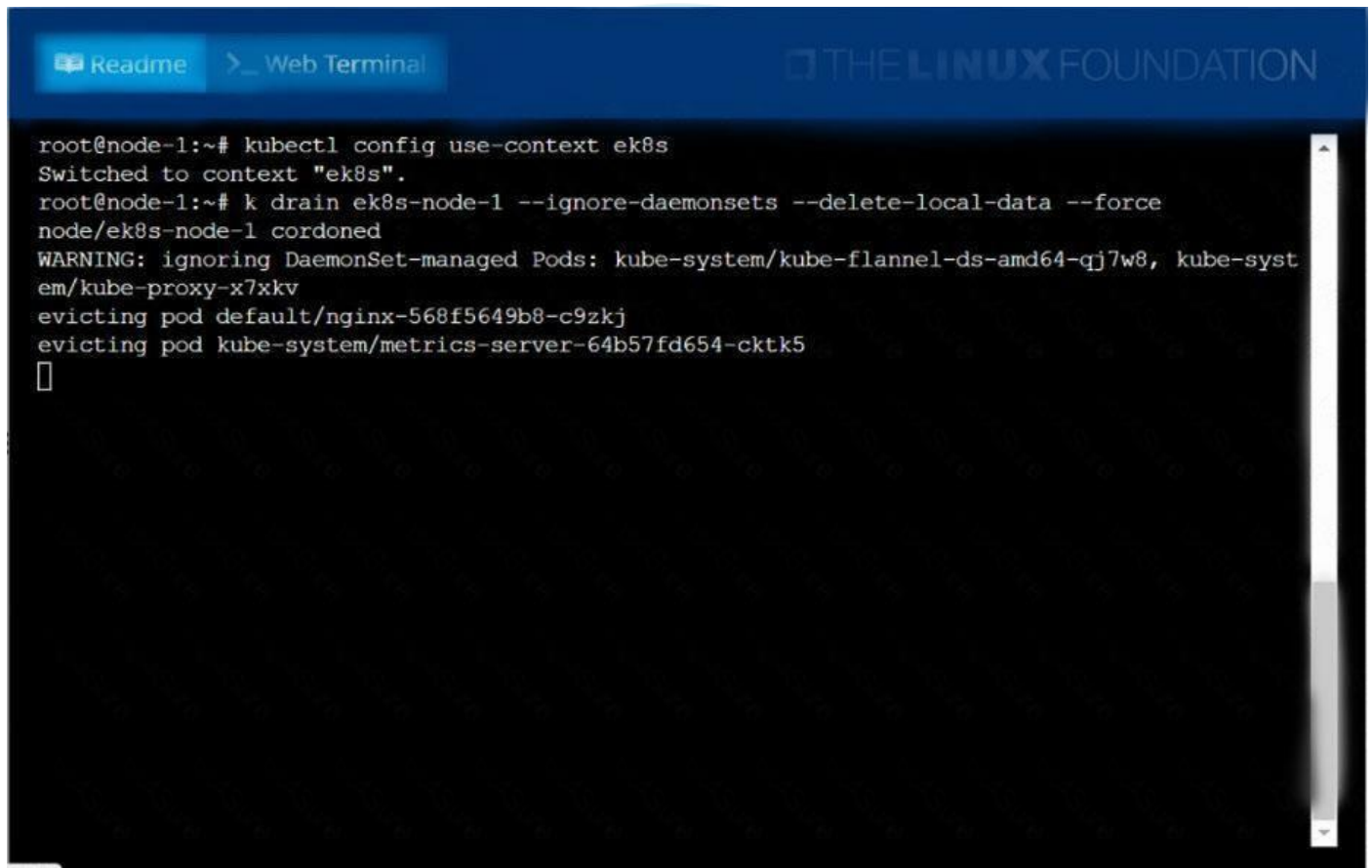
**Answer: A**

**Explanation:**

rrrect.  
See the solution below.

Explanation:

solution



The screenshot shows a terminal window with a dark background and light blue text. At the top, there is a blue header bar with the text "THE LINUX FOUNDATION" on the right and two buttons on the left: "Readme" and "Web Terminal". The terminal content shows the following commands and output:

```
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
█
```

## Question 8

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

### Options:

A. Option is correct.  
See the solution below.

Explanation:

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

**Answer: A**

**Explanation:**

rrrect.

See the solution below.

Explanation:

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

### Question 9

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

**Options:**

A. Option is correct.

See the solution below.

Explanation:

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

**Answer: A**

**Explanation:**

rrrect.

See the solution below.

Explanation:

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

### Question 10

Create an nginx pod with container Port 80 and it should only receive traffic only it checks the endpoint / on port 80 and verify and delete the pod.

### Options:

A. `kubectl run nginx --image=nginx --restart=Never --port=80 --`

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

`// add the readinessProbe section and create`

`vim nginx-pod.yaml`

`run: nginx`

`name: nginx`

`spec:`

`containers:`

`- image: nginx`

`name: nginx`

`ports:`

`- containerPort: 60`

`readinessProbe:`

`httpGet:`

`path: /`

`port: 60`

`restartPolicy: Never`

`kubectl apply -f nginx-pod.yaml`

`// verify`

`kubectl describe pod nginx | grep -i readiness`

`kubectl delete po nginx`

B. `kubectl run nginx --image=nginx --restart=Never --port=80 --`

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

`// add the readinessProbe section and create`

`vim nginx-pod.yaml`

`apiVersion: v1`

`kind: Pod`

`metadata:`

`labels:`

`run: nginx`

`name: nginx`

`spec:`

containers:

- image: nginx

name: nginx

ports:

- containerPort: 80

readinessProbe:

httpGet:

path: /

port: 80

restartPolicy: Never

kubectl apply -f nginx-pod.yaml

// verify

kubectl describe pod nginx | grep -i readiness

kubectl delete po nginx

**Answer: B**

**Would you like to see more? Don't miss our CKA PDF file at:**

<https://www.certification-questions.com/linux-foundation-pdf/cka-pdf.html>