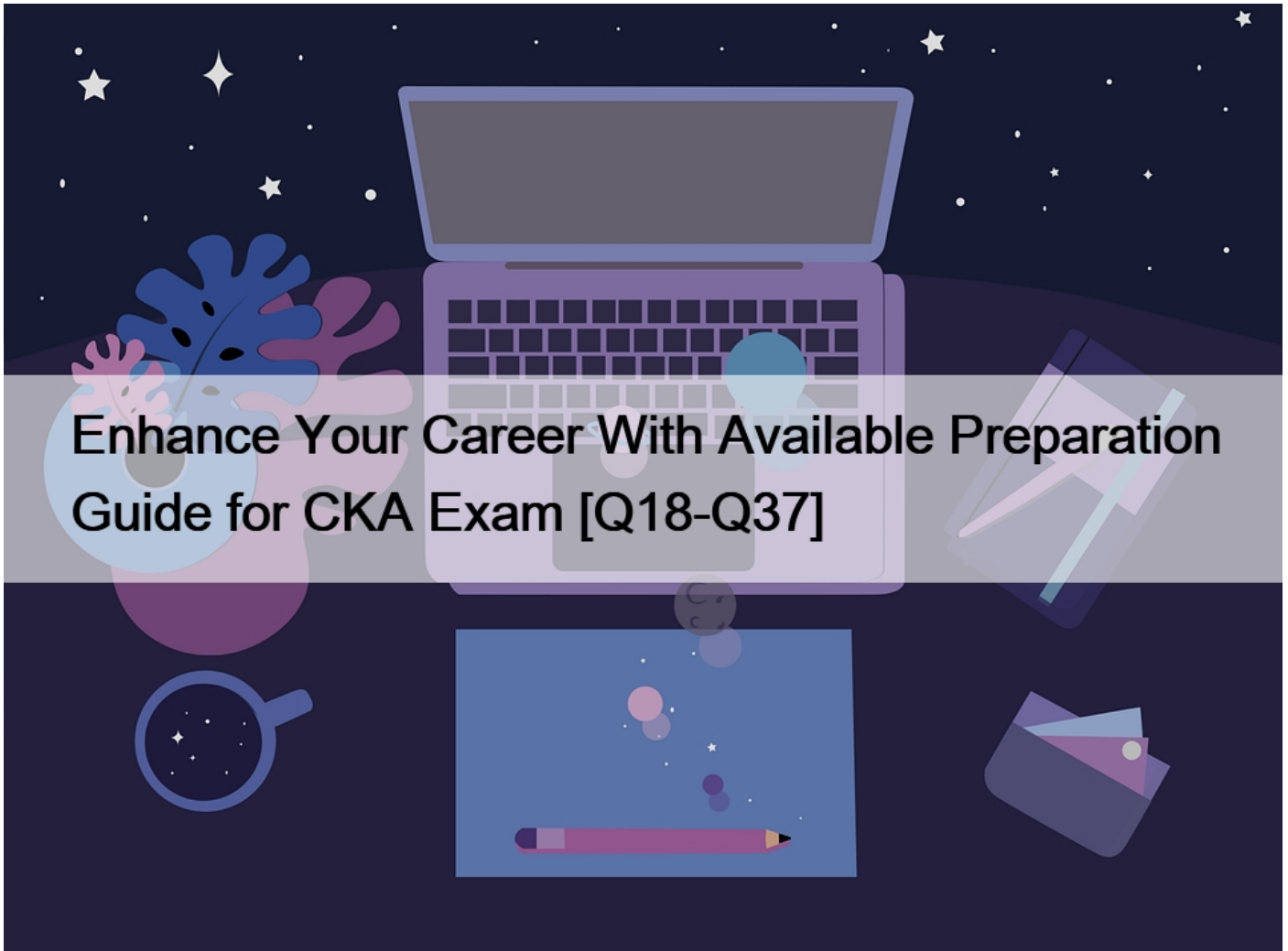


## Enhance Your Career With Available Preparation Guide for CKA Exam [Q18-Q37]



### Enhance Your Career With Available Preparation Guide for CKA Exam Get Special Discount Offer of CKA Certification Exam Sample Questions and Answers NEW QUESTION 18

Create the nginx pod with version 1.17.4 and expose it on port 80  
kubectl run nginx --image=nginx:1.17.4 --restart=Never --port=80

#### NEW QUESTION 19

List all the events sorted by timestamp and put them into file.log and verify

```
* kubectl get events --sort-by=.metadata.creationTimestamp
```

```
kubectl get events --sort-by=.metadata.creationTimestamp >
```

```
test-file.log
```

```
cat test-file.log
* kubectl get events &#8211;sort-by=.metadata.creationTimestamp

// putting them into file.log

kubectl get events &#8211;sort-by=.metadata.creationTimestamp >

cat test-file.log
* kubectl get events &#8211;sort-by=.metadata.creationTimestamp

// putting them into file.log

kubectl get events &#8211;sort-by=.metadata.creationTimestamp >

test-file.log

cat test-file.log
```

## NEW QUESTION 20

Score: 4%



Task

Schedule a pod as follows:

\* Name: nginx-kusc00401

\* Image: nginx

\* Node selector: disk=ssd

Solution:

```
#yaml
```

```
apiVersion: v1
```

```
kind: Pod
```

metadata:

name: nginx-kusc00401

spec:

containers:

&#8211; name: nginx

image: nginx

imagePullPolicy: IfNotPresent

nodeSelector:

disk: spinning

#

kubectl create -f node-select.yaml

## NEW QUESTION 21

Score:7%



### Context

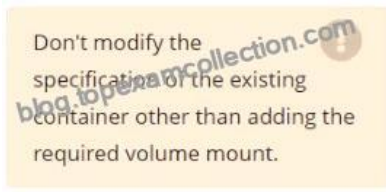
An existing Pod needs to be integrated into the Kubernetes built-in logging architecture (e. g. kubectl logs). Adding a streaming sidecar container is a good and common way to accomplish this requirement.

### 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.



Solution:

```
#  
  
kubectl get pod big-corp-app -o yaml  
  
#  
  
apiVersion: v1  
  
kind: Pod  
  
metadata:  
  
name: big-corp-app  
  
spec:  
  
containers:  
  
&#8211; name: big-corp-app  
  
image: busybox  
  
args:  
  
&#8211; /bin/sh  
  
&#8211; -c  
  
&#8211; >  
  
i=0;  
  
while true;  
  
do  
  
echo &#8220;$(date) INFO $i&#8221; >> /var/log/big-corp-app.log;
```

```
i=$((i+1));  
  
sleep 1;  
  
done  
  
volumeMounts:  
  
&#8211; name: logs  
  
mountPath: /var/log  
  
&#8211; name: count-log-1  
  
image: busybox  
  
args: [/bin/sh, -c, &#8216;tail -n+1 -f /var/log/big-corp-app.log&#8217;]  
  
volumeMounts:  
  
&#8211; name: logs  
  
mountPath: /var/log  
  
volumes:  
  
&#8211; name: logs  
  
emptyDir: {  
  
}  
  
#  
  
kubectl logs big-corp-app -c count-log-1
```

## NEW QUESTION 22

Check the image version in pod without the describe command  
See the solution below.

Explanation

```
kubectl get po nginx -o  
  
jsonpath='{.spec.containers[].image}'&#8220;n&#8221;}&#8217;
```

## NEW QUESTION 23

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

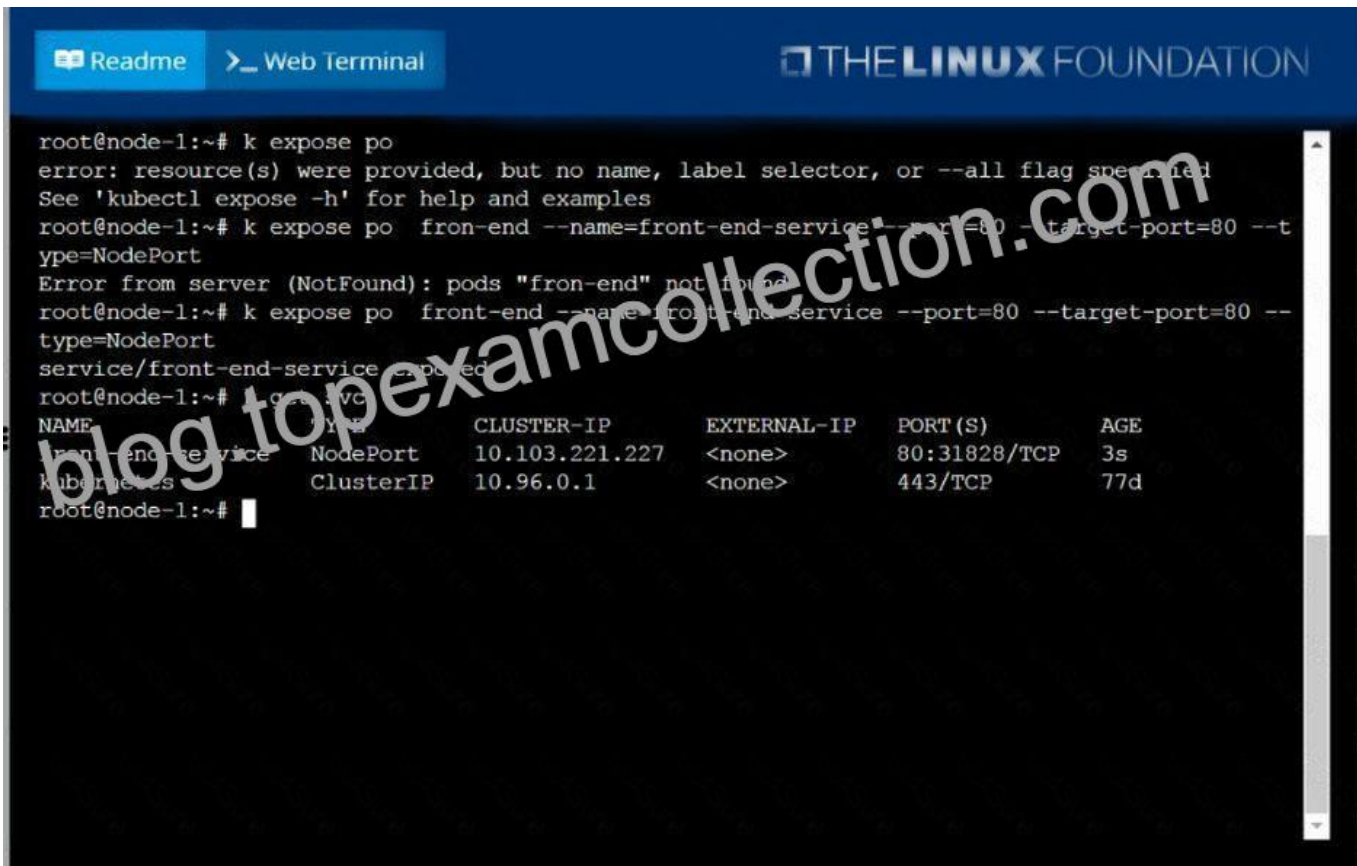
front-end.

See the solution below.

Explanation

solution

F:\WorkData Entry\WorkData Entry\20200827CKA8 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 "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:~#
```

### NEW QUESTION 24

Change the label for one of the pod to env=uat and list all the pods to verify

kubectl label pod/nginx-dev3 env=uat &#8211;overwrite kubectl get pods &#8211;show-labels

### NEW QUESTION 25

List pod logs named &#8220;frontend&#8221; and search for the pattern &#8220;started&#8221; and write it to a file &#8220;/opt/error-logs&#8221;

Kubectl logs frontend | grep -i &#8220;started&#8221; > /opt/error-logs

### NEW QUESTION 26

Label a node as app=test and verify

kubectl label node node-name app=test // Verify kubectl get no -show-labels kubectl get no -l app=test

### NEW QUESTION 27

Undo the deployment with the previous version and verify

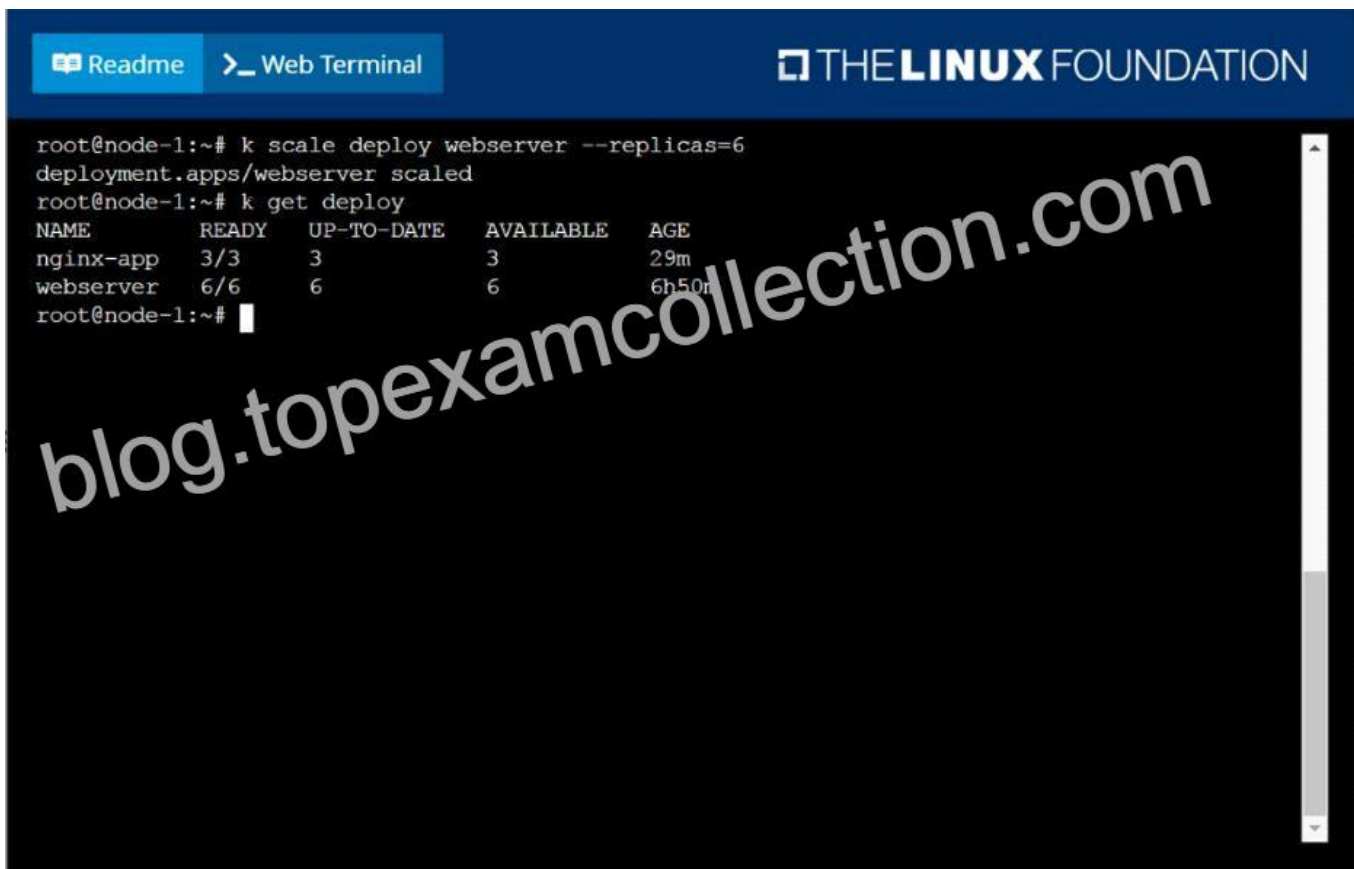
everything is Ok

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

### NEW QUESTION 28

Scale the deployment webserver to 6 pods.

solution



The screenshot shows a terminal window with a dark background. At the top, there are two buttons: 'Readme' and 'Web Terminal'. To the right, the logo for 'THE LINUX FOUNDATION' is visible. The terminal output shows the following commands and results:

```
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

The terminal ends with the prompt `root@node-1:~#`. A large watermark `blog.topexamcollection.com` is overlaid diagonally across the terminal output.

### NEW QUESTION 29

Create a configmap called myconfigmap with literal value

```
appname=myapp
```

```
* kubectl create cm myconfigmap --from-literal=appname=myapp
```

```
// Verify
```

```
kubectl get cm -o yaml
```

(or)

```
kubectl describe cm
```

```
* kubectl create cm myconfigmap --from-literal=appname=myapp
```

```
// Verify
```

(or)

```
kubectl describe cm
```

### NEW QUESTION 30

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

```
kubectl run busybox --image=busybox --rm --restart=Never --
```

```
/bin/sh -c "echo hello world";
```

```
kubectl get po # You shouldn't see pod with the name busybox;
```

### NEW QUESTION 31

Get the list of pods of webapp deployment

```
* // Get the label of the deployment
```

```
kubectl get deploy --show-labels
```

```
kubectl get pods -l app=webapp
```

```
* // Get the label of the deployment
```

```
kubectl get deploy --show-labels
```

```
// Get the pods with that label
```

```
kubectl get pods -l app=webapp
```

### NEW QUESTION 32

Schedule a pod as follows:

Name: nginx-kusc00101

Image: nginx

Node selector: disk=ssd

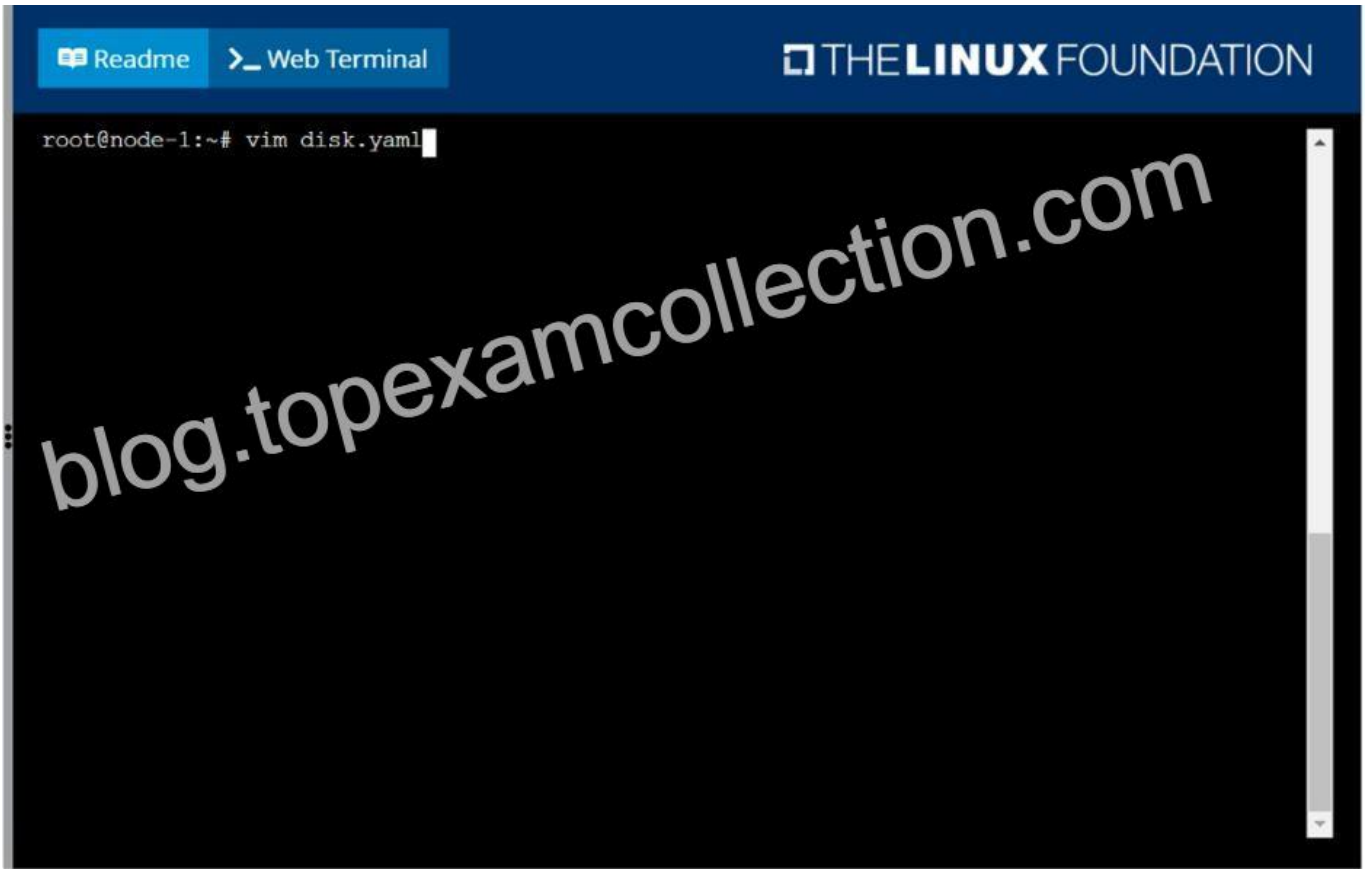
See the solution below.

Explanation



solution

F:\WorkData Entry\WorkData Entry\20200827CKA6 B.JPG



F:\WorkData Entry\WorkData Entry\20200827CKA6 C.JPG

```
Readme Web Terminal THE LINUX FOUNDATION

apiVersion: v1
kind: Pod
metadata:
  name: nginx-kusc00101
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  nodeSelector:
    disk: ssd

"disk.yaml" [New] 11L, 176C written
```

F:\WorkData Entry\WorkData Entry\20200827CKA6 D.JPG

```
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           13m
ds-kusc00201-hzr19g                 1/1     Running   0           13m
foo                                  1/1     Running   0           6h1m
prnt-enc                             1/1     Running   0           6h1m
hungry-bear                          1/1     Running   0           9m37s
kucc8                                  3/3     Running   0           7m37s
nginx-kusc00101                     1/1     Running   0           9s
webserver-84c55967f4-qzjcv          1/1     Running   0           6h16m
webserver-84c55967f4-t479l          1/1     Running   0           6h16m
root@node-1:~#
```

### NEW QUESTION 33

Add a taint to node `worker-2`; with effect as `NoSchedule`; and

list the node with taint effect as `NoSchedule`;

```
* // Add taint to node worker-2;
```

```
kubectl taint nodes worker-2 key=value:NoSchedule
```

```
.items[*] { .metadata.name } { .spec.taints[(
```

```
@.effect==NoSchedule; )].effect} {n; } {end} | awk NF==2
```

```
{print $0};
```

```
* // Add taint to node worker-2;
```

```
kubectl taint nodes worker-2 key=value:NoSchedule
```

```
// Verify
```

```
// Using custom-columns, you can customize which column to
```

```
be printed
```

```
kubectl get nodes -o customcolumns=NAME:.metadata.name,TAINTS:.spec.taints ;no-headers
```

```
// Using jsonpath
```

```
kubectl get nodes -o jsonpath={range
```

```
.items[*] { .metadata.name } { .spec.taints[(
```

```
@.effect==NoSchedule; )].effect} {n; } {end} | awk NF==2
```

```
{print $0};
```

### NEW QUESTION 34

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.

Solution:

Readme Web Terminal THE **LINUX** FOUNDATION

```
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
```

blog.topexamcollection.com

Readme Web Terminal THE **LINUX** FOUNDATION

```
apiVersion: v1  
kind: Pod  
metadata:  
  name: busybox1  
  labels:  
    name: busybox  
spec:  
  containers:  
  - image: busybox:1.28  
    command:  
    - sleep  
    - 3600  
  dnsPolicy: ClusterFirst  
  restartPolicy: Never
```

blog.topexamcollection.com

The screenshot shows a terminal window with the following content:

```

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
root@node-1:~# k create -f dns.yaml
pod/busybox1 created
root@node-1:~# k get po -o wide | grep nginx-random
nginx-random-6d5766bbdc-ptb 2/2 Running 0 103s 10.244.2.16 k8s-node-1 <none>
root@node-1:~# k exec -it busybox1 -- nslookup nginx-random
Server: 10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name:      nginx-random
Address 1: 10.111.37.132 nginx-random.default.svc.cluster.local
root@node-1:~# k exec -it busybox1 -- nslookup nginx-random > /opt/KUNW00601/service.dns
root@node-1:~# k exec -it busybox1 -- nslookup 10-244-2-16.default.pod
Server: 10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name:      10-244-2-16.default.pod
Address 1: 10.244.2.16 10-244-2-16.nginx-random.default.svc.cluster.local
root@node-1:~# k exec -it busybox1 -- nslookup 10-244-2-16.default.pod > /opt/KUNW00601/pod.dns

```

**NEW QUESTION 35**

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

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

# then

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

# or

```
kubectl describe po nginx | grep value1
```

```
* 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
```

### NEW QUESTION 36

Get all the pods with label `env`;

```
kubectl get pods -L env
```

### NEW QUESTION 37

Create an nginx pod and load environment values from the above configmap `keyvalcfgmap`; and exec into the pod and verify the environment variables and delete the pod

\* // first run this command to save the pod yml

```
kubectl run nginx --image=nginx --restart=Always --dry-run -o
```

```
yml > nginx-pod.yml
```

// edit the yml to below file and create

```
vim nginx-pod.yml
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
labels:
```

```
run: nginx
```

```
name: nginx
```

```
spec:
```

```
containers:
```

```
  -- image: nginx
```

```
  name: nginx
```

```
  envFrom:
```

```
  -- configMapRef:
```

```
    name: keyvalcfgmap
```

```
  restartPolicy: Always
```

```
kubectl apply -f nginx-pod.yml
```

```
// verify
```

```
kubectl exec -it nginx &#8212; env
```

```
kubectl delete po nginx
```

```
* // first run this command to save the pod yaml
```

```
kubectl run nginx &#8211;image=nginx &#8211;restart=Always &#8211;dry-run -o
```

```
yaml > nginx-pod.yml
```

```
// edit the yaml to below file and create
```

```
vim nginx-pod.yml
```

```
apiVersion: v1
```

```
name: nginx
```

```
envFrom:
```

```
&#8211; configMapRef:
```

```
name: keyvalcfgmap
```

```
restartPolicy: Always
```

```
kubectl apply -f nginx-pod.yml
```

```
// verify
```

```
kubectl exec -it nginx &#8212; env
```

```
kubectl delete po nginx
```

The Linux Foundation Certified Kubernetes Administrator (CKA) Program Certification Exam is designed to test the skills and knowledge of individuals in the field of Kubernetes administration. The exam is aimed at individuals who are responsible for

designing, deploying, and maintaining Kubernetes clusters in production environments. The CKA Program Certification Exam is a performance-based exam, which means that candidates will be required to perform tasks on a live Kubernetes cluster rather than answering multiple-choice questions.

**Updated CKA Dumps Questions Are Available For Passing Linux Foundation Exam:**

<https://www.topexamcollection.com/CKA-vce-collection.html>