

## CKA Dumps

### Certified Kubernetes Administrator (CKA) Program

<https://www.certleader.com/CKA-dumps.html>



#### NEW QUESTION 1

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

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

```
kubectrl run nginx --image=nginx --restart=Never --env=var1=value1
# then
kubectrl exec -it nginx -- env
# or
kubectrl exec -it nginx -- sh -c 'echo $var1'
# or
kubectrl describe po nginx | grep value1
```

#### NEW QUESTION 2

Monitor the logs of pod foo and:

Extract log lines corresponding to error unable-to-access-website  
Write them to /opt/KULM00201/foo

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

```
solution
F:\Work\Data Entry Work\Data Entry\20200827\CKA\1 B.JPG

F:\Work\Data Entry Work\Data Entry\20200827\CKA\1 C.JPG
```

### NEW QUESTION 3

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.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution

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

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

### NEW QUESTION 4

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

- A. Mastered  
B. Not Mastered

**Answer:** A

**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 ?C
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
```

**NEW QUESTION 5**

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.

- A. Mastered  
B. Not Mastered

**Answer:** A

**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 app-data, with access mode ReadWriteMany, storage class name shared, 2Gi of storage capacity and the host path /srv/app-data.

\* 2. Save the file and create the persistent volume. Image for post

\* 3. View the persistent volume.

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: PersistentVolume apiVersion: v1 metadata: name: app-data spec:

accessModes: - ReadWriteMany resources:

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

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

Image for post

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: config pvc: persistentVolumeClaim: claimName: app-data containers: - image: nginx name: app volumeMounts: - mountPath: "/srv/app-data" name: config pvc
```

**NEW QUESTION 6**

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.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\2 C.JPG

**NEW QUESTION 7**

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

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

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

**NEW QUESTION 8**

Get list of all pods in all namespaces and write it to file `??/opt/pods-list.yaml??`

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

kubectrl get po ?Call-namespaces > /opt/pods-list.yaml

**NEW QUESTION 9**

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

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

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

**NEW QUESTION 10**

Create a file:

/opt/KUCC00302/kucc00302.txt that lists all pods that implement service baz in namespace development.

The format of the file should be one pod name per line.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\11 B.JPG

F:\Work\Data Entry Work\Data Entry\20200827\CKA\11 C.JPG

F:\Work\Data Entry Work\Data Entry\20200827\CKA\11 D.JPG

**NEW QUESTION 10**

Check the image version in pod without the describe command

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

```
kubectl get po nginx -o jsonpath='{.spec.containers[].image}'
```

**NEW QUESTION 12**

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.

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

kubectl run --generator=run-pod/v1 --image=nginx -- labels=env=prod nginx-prod --dry-run -o yaml > nginx-prod-pod.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
```

**NEW QUESTION 14**

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 http://nginx.org/images/nginx-1.15.10-1.el7.ngx.el7.centos.x86\_64.rpm. Any spec files required should be placed in the /etc/kubernetes/manifests directory on the node.

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 ?Ci
```

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution

F:\Work\Data Entry Work\Data Entry\20200827\CKA\21 C.JPG

F:\Work\Data Entry Work\Data Entry\20200827\CKA\21 D.JPG

F:\Work\Data Entry Work\Data Entry\20200827\CKA\21 E.JPG

F:\Work\Data Entry Work\Data Entry\20200827\CKA\21 F.JPG

F:\Work\Data Entry Work\Data Entry\20200827\CKA\21 G.JPG

**NEW QUESTION 16**

Check the Image version of nginx-dev pod using jsonpath

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

```
kubectl get po nginx-dev -o jsonpath='{.spec.containers[0].image}'
```

**NEW QUESTION 21**

Create a busybox pod and add --sleep 3600 command

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

kubectrl run busybox --image=busybox --restart=Never -- /bin/sh -c "sleep 3600"

**NEW QUESTION 26**

List all the pods sorted by name

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

kubectrl get pods --sort-by=.metadata.name

**NEW QUESTION 28**

For this item, you will have to ssh to the node `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.

- A. Mastered
- B. Not Mastered

**Answer:** A

**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 `apt` has been configured so that you can install the required tools.

**NEW QUESTION 31**

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`

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution  
F:\Work\Data Entry Work\Data Entry\20200827\CKA\18 C.JPG

**NEW QUESTION 33**

Schedule a pod as follows:

Name: nginx-kusc00101  
Image: nginx  
Node selector: disk=ssd

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution  
F:\Work\Data Entry Work\Data Entry\20200827\CKA\6 B.JPG

F:\Work\Data Entry Work\Data Entry\20200827\CKA\6 C.JPG

F:\Work\Data Entry Work\Data Entry\20200827\CKA\6 D.JPG

**NEW QUESTION 35**

Create an nginx pod and list the pod with different levels of verbosity

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

```
// create a pod
kubectl run nginx --image=nginx --restart=Never --port=80
// List the pod with different verbosity kubectl get po nginx --v=7
kubectl get po nginx --v=8 kubectl get po nginx --v=9
```

**NEW QUESTION 38**

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

- A. Mastered
- B. Not Mastered

**Answer:** A

**Explanation:**

solution  
F:\Work\Data Entry Work\Data Entry\20200827\CKA\15 B.JPG

F:\Work\Data Entry Work\Data Entry\20200827\CKA\15 C.JPG

**NEW QUESTION 42**

.....

## Thank You for Trying Our Product

\* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

\* One year free update

You can enjoy free update one year. 24x7 online support.

\* Trusted by Millions

We currently serve more than 30,000,000 customers.

\* Shop Securely

All transactions are protected by VeriSign!

**100% Pass Your CKA Exam with Our Prep Materials Via below:**

<https://www.certleader.com/CKA-dumps.html>