

CKA Dumps

Certified Kubernetes Administrator (CKA) Program

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



NEW QUESTION 1

Create a deployment as follows:

- > Name:nginx-random
- > Exposed via a servicenginx-random
- > Ensure that the service & podare accessible via theirrespective DNS records
- > The container(s) within anypod(s) running as a part of thisdeployment should use thenginxImage

Next, use the utilitynslookupto lookup the DNS records of the service &pod and write the output to /opt/KUNW00601/service.dnsand/opt/KUNW00601/pod.dnsrespectively.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Solution:

F:\Work\Data Entry Work\Data Entry\20200827\CKA\17 C.JPG

```

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.yam
    
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\17 D.JPG

```

apiVersion: v1
kind: Pod
metadata:
  name: busybox1
  labels:
    name: busybox
spec:
  containers:
  - image: busybox:1.28
    command:
      - sleep
      - "3600"
    name: busybox
    
```

F:\Work\Data Entry Work\Data Entry\20200827\CKA\17 E.JPG

```

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-ptzv2 1/1 Running 0 103s 10.244.2.16 k8s-node-1 <none> <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 2

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

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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 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

```

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

```

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

```

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 4

List the nginx pod with custom columns POD_NAME and POD_STATUS

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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

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:
 storage: 2Gi
 accessModes:
 - ReadWriteMany
 hostPath:
 path: /srv/app-data
 storageClassName: shared

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.

```
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  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: PersistentVolumeClaim
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

```
njerry191@cloudshell:~ (extreme-clone-265411)$ 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-265411)$ 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: config-pvc
    persistentVolumeClaim:
      claimName: app-data
containers:
- image: nginx
  name: app
  volumeMounts:
  - mountPath: "/srv/app-data"
    name: config-pvc
```

NEW QUESTION 6

List all the pods sorted by name

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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

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:

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

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 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@wk8s-node-0] $ | sudo -i
```

- A. Mastered
- B. Not Mastered

Answer: A

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 '18.04.5 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
```

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

```

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 10.250.5.34 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 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

From the pod labelname=cpu-utilizer, find podsrunning high CPU workloads and write the name of the pod consumingmost CPU to thefile/opt/KUTR00102/KUTR00102.txt(which already exists).

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

solution
F:\Work\Data Entry Work\Data Entry\20200827\CKA\16 B.JPG

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

```

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:~#
    
```

NEW QUESTION 22

Scale the deploymentwebserverto6pods.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

solution
F:\Work\Data Entry Work\Data Entry\20200827\CKA\14 B.JPG

```

root@node-1:~# k scale deployment/webserver --replicas=6
deployment.apps/webserver scaled
root@node-1:~# k get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-app     3/3     3             3           29m
webserver     6/6     6             6           6h50m
root@node-1:~#
    
```

NEW QUESTION 27

Create a pod that having 3 containers in it? (Multi-Container)

- A. Mastered

B. Not Mastered

Answer: A

Explanation:

image=nginx, image=redis, image=consul Name nginx container as ??nginx-container?? Name redis container as ??redis-container?? Name consul container as ??consul-container??

Create a pod manifest file for a container and append container section for rest of the images

kubectl run multi-container --generator=run-pod/v1 --image=nginx -- dry-run -o yaml > multi-container.yaml

then

vim multi-container.yaml apiVersion: v1

kind: Pod metadata: labels:

run: multi-container name: multi-container spec:

containers:

- image: nginx

name: nginx-container

- image: redis

name: redis-container

- image: consul

name: consul-container

restartPolicy: Always

NEW QUESTION 29

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

A. Mastered

B. Not Mastered

Answer: A

Explanation:

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

NEW QUESTION 30

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

A. Mastered

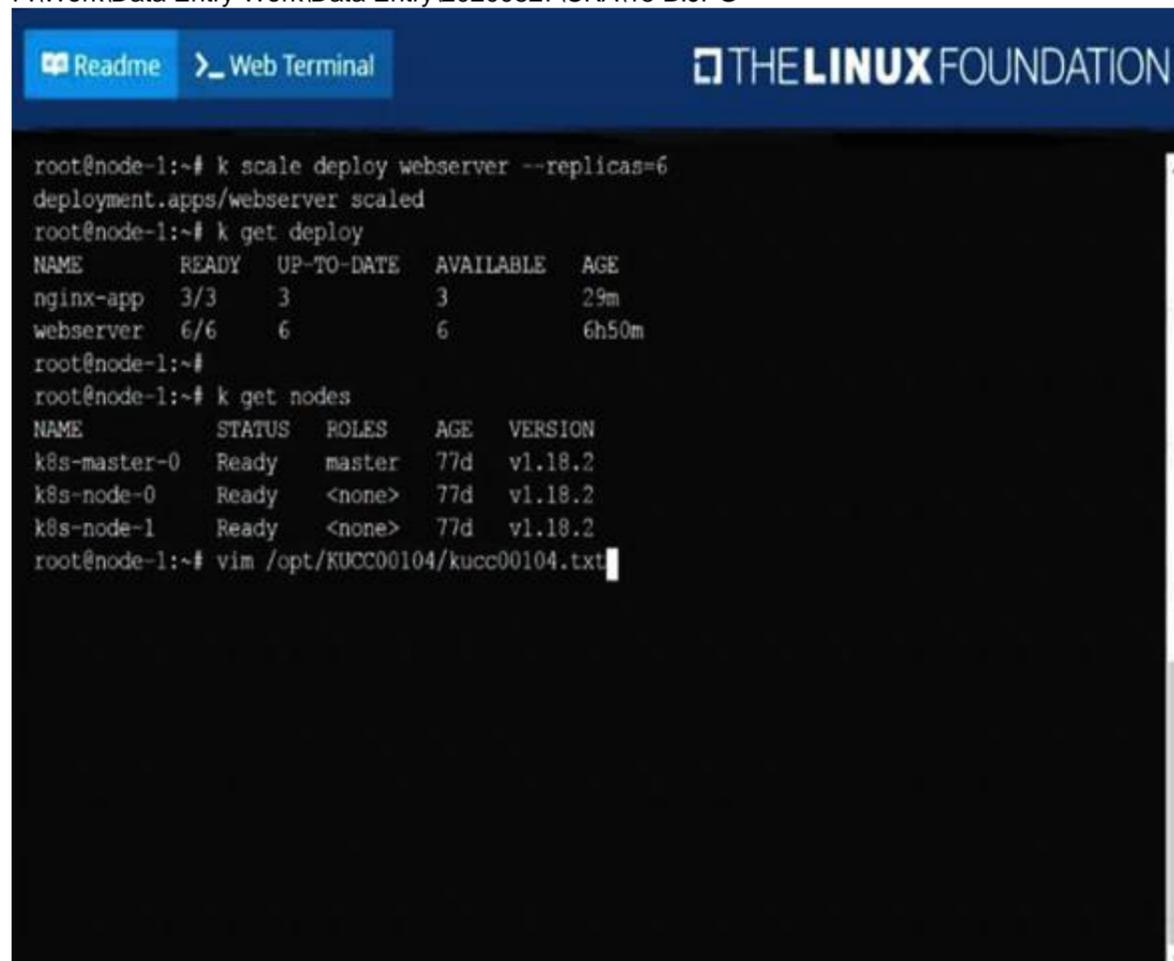
B. Not Mastered

Answer: A

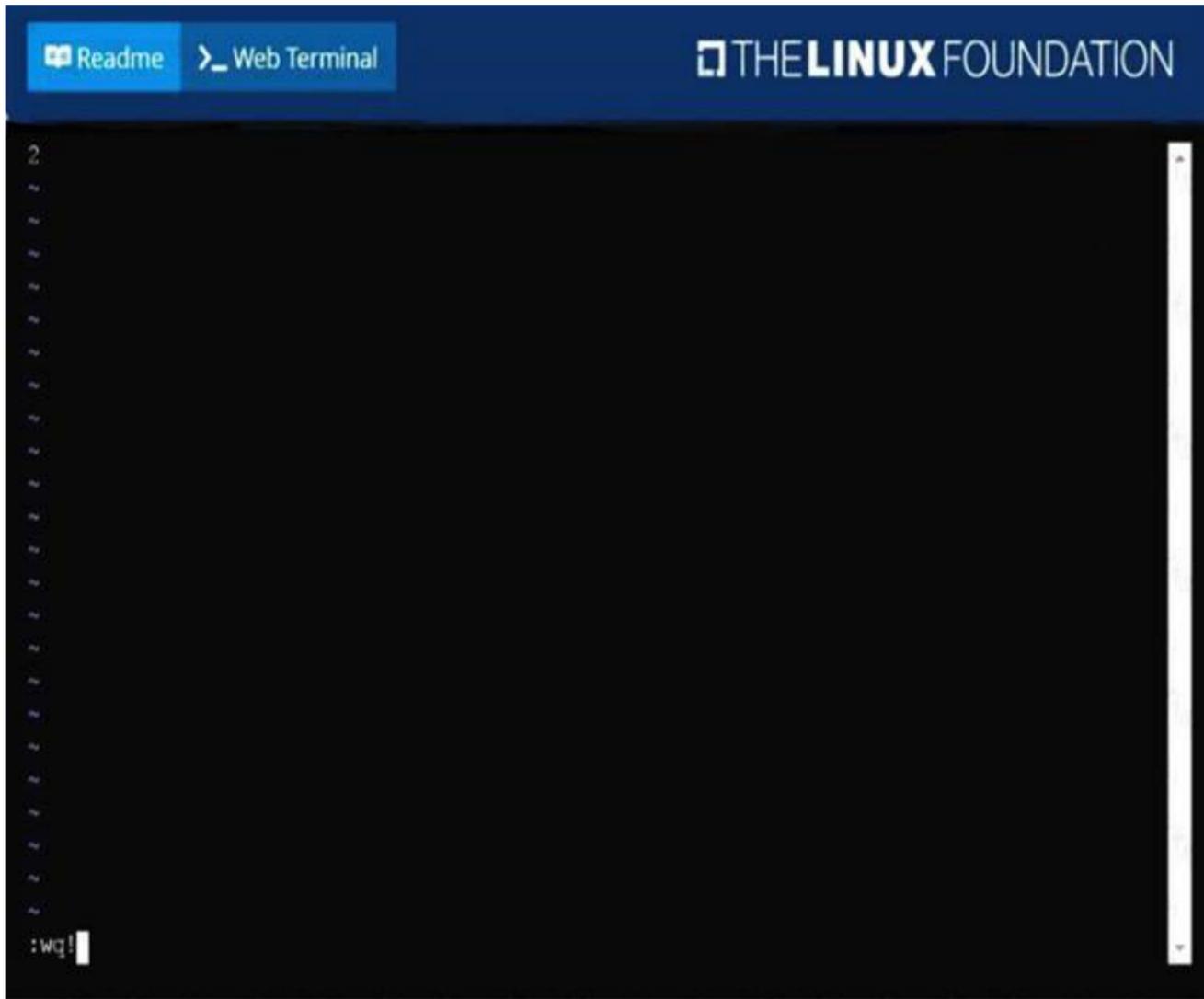
Explanation:

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 35

.....

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>