



CKS^{Q&As}

Certified Kubernetes Security Specialist (CKS) Exam

Pass Linux Foundation CKS Exam with 100% Guarantee

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

<https://www.pass4itsure.com/cks.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

You can switch the cluster/configuration context using the following command:

```
[desk@cli] $ kubectl config use-context qa
```

Context:

A pod fails to run because of an incorrectly specified ServiceAccount

Task:

Create a new service account named backend-qa in an existing namespace qa, which must not have access to any secret.

Edit the frontend pod yaml to use backend-qa service account

Note: You can find the frontend pod yaml at /home/cert_masters/frontend-pod.yaml

A. See the explanation below

B. Placeholder

Correct Answer: A

```
[desk@cli] $ k create sa backend-qa -n qasa/backend-qa created [desk@cli] $ k get role,rolebinding -n qaNo resources found in qa namespace. [desk@cli] $ k create role backend -n qa --resource pods,namespaces,configmaps --verb list# No access to secret [desk@cli] $ k create rolebinding backend -n qa --role backend --serviceaccount qa:backend-qa [desk@cli] $ vim /home/ cert_masters/frontend-pod.yaml uk.co.certification.simulator.questionpool.PList@120e0660 [desk@cli] $ k apply -f /home/cert_masters/frontend-pod.yamlpod created [desk@cli] $ k create sa backend-qa -n qaserviceaccount/backend-qa created [desk@cli] $ k get role,rolebinding -n qaNo resources found in qa namespace. [desk@cli] $ k create role backend -n qa --resource pods,namespaces,configmaps --verb listrole.rbac.authorization.k8s.io/backend created [desk@cli] $ k create rolebinding backend -n qa --role backend --serviceaccount qa:backendqarolebinding.rbac.authorization.k8s.io/backend created [desk@cli] $ vim /home/cert_masters/frontend-pod.yaml apiVersion: v1 kind: Pod metadata: name: frontend spec: serviceAccountName: backend-qa # Add this image: nginx name: frontend [desk@cli] $ k apply -f /home/cert_masters/frontend-pod.yamlpod/frontend createdhttps://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/
```

QUESTION 2



```
candidate@cli:~$ kubectl config use-context KSCS00101
Switched to context "KSCS00101".
candidate@cli:~$ cat /home/candidate/KSCS00101/network-policy.yaml
---
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: ""
  namespace: ""
spec:
  podSelector: {}
  policyTypes: []
candidate@cli:~$ vim /home/candidate/KSCS00101/network-policy.yaml
candidate@cli:~$
```

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: "defaultdeny"
  namespace: "testing"
spec:
  podSelector: {}
  policyTypes:
  - Egress
  egress:
  - to:
    - podSelector: {}
      namespaceSelector:
        matchLabels:
          access: testingproject
```

```
candidate@cli:~$ vim /home/candidate/KSCS00101/network-policy.yaml
candidate@cli:~$ vim /home/candidate/KSCS00101/network-policy.yaml
candidate@cli:~$ kubectl label ns testing access=testingproject
namespace/testing labeled
candidate@cli:~$ cat /home/candidate/KSCS00101/network-policy.yaml
---
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: "defaultdeny"
  namespace: "testing"
spec:
  podSelector: {}
  policyTypes:
  - Egress
  egress:
  - to:
    - podSelector: {}
      namespaceSelector:
        matchLabels:
          access: testingproject
candidate@cli:~$ kubectl create -f /home/candidate/KSCS00101/network-policy.yaml
networkpolicy.networking.k8s.io/defaultdeny created
candidate@cli:~$ kubectl -n testing describe networkpolicy
Name:          defaultdeny
Namespace:     testing
Created on:    2022-05-20 14:28:27 +0000 UTC
Labels:        <none>
Annotations:   <none>
Spec:
  PodSelector:    <none> (Allowing the specific traffic to all pods in this namespace)
  Not affecting ingress traffic
  Allowing egress traffic:
    To Port: <any> (traffic allowed to all ports)
    To:
      NamespaceSelector: access=testingproject
      PodSelector: <none>
  Policy Types: Egress
candidate@cli:~$
```



Create a RuntimeClass named gvisor-rc using the prepared runtime handler named runsc.

Create a Pods of image Nginx in the Namespace server to run on the gVisor runtime class

A. See the explanation below:

B. Placeholder

Correct Answer: A

Install the Runtime Class for gVisor

```
{ # Step 1: Install a RuntimeClass cat evt.type in (open,openat,creat) and evt.is_open_exec=true and container and not
runc_writing_exec_fifo and not runc_writing_var_lib_docker and not user_known_container_drift_activities and
evt.rawres>=0 output: > %evt.time,%user.uid,%proc.name # Add this/Refer falco documentation priority: ERROR
[node01@cli] $ vim /etc/falco/falco.yaml
```

[CKS VCE Dumps](#)

[CKS Exam Questions](#)

[CKS Braindumps](#)