Valid CKS Exam Experience, Pdf CKS Pass Leader

Linux Foundation Certified Kubernetes Security Specialist (CKS) Sample Questions (Q30-Q35):

NEW QUESTION # 30

Answer: A

NEW QUESTION # 31

Answer: A

NEW QUESTION # 32

P.S. Free 2026 Linux Foundation CKS dumps are available on Google Drive shared by DumpsFree: https://drive.google.com/open?id=1TNH7LOyiD6fpuBYNnOj7j1LSEWPonaU3

You have to know that a choice may affect your very long life. Our CKS guide quiz is willing to provide you with a basis for making judgments. You can download the trial version of our CKS practice prep first. After using it, you may have a better understanding of some of the advantages of CKS Exam Materials. We have three versions of our CKS learning quiz: the PDF, Software and APP online for you to choose.

The CKS certification exam covers a wide range of topics, including Kubernetes cluster architecture, network security, container security, access management, and security auditing. CKS exam is designed to assess the candidate's knowledge of security best practices, as well as their ability to implement and manage security controls effectively. Certified Kubernetes Security Specialist (CKS) certification exam is vendor-neutral, which means that it is not tied to any particular technology or platform, and is recognized by organizations worldwide.

Linux Foundation has recently announced the launch of a new certification exam – the Certified Kubernetes Security Specialist (CKS). CKS Exam is designed to assess and validate the skills and knowledge of IT professionals who specialize in securing Kubernetes clusters.

>> Valid CKS Exam Experience <<

Free PDF Quiz 2026 CKS: Certified Kubernetes Security Specialist (CKS) –

Reliable Valid Exam Experience

Certified Kubernetes Security Specialist (CKS) CKS study guide are high quality, since we have a professional team to collect the information for the exam, and we can ensure you that CKS study guide you receive are the latest information we have. In order to strengthen your confidence for Linux Foundation CKS Exam Dumps, we are pass guarantee and money back guarantee.

The CKS Certification is a valuable credential for IT professionals who work with Kubernetes and containerized applications. It demonstrates a candidate's commitment to maintaining the highest standards of security in their work and provides a competitive edge in the job market. Certified Kubernetes Security Specialist (CKS) certification exam is rigorous and challenging, requiring candidates to have a strong understanding of Kubernetes security best practices. However, it is also a rewarding experience, as successful candidates will have the skills and knowledge to secure Kubernetes environments and protect their organizations from cyber threats.

Linux Foundation Certified Kubernetes Security Specialist (CKS) Sample Questions (Q13-Q18):

NEW QUESTION # 13
Create a PSP that will only allow the persistentvolumeclaim as the volume type in the namespace restricted.
Create a new PodSecurityPolicy named prevent-volume-policy which prevents the pods which is having different volumes mount apart from persistentvolumeclaim.
Create a new ServiceAccount named psp-sa in the namespace restricted.
Create a new ClusterRole named psp-role, which uses the newly created Pod Security Policy prevent-volume-policy Create a new ClusterRoleBinding named psp-role-binding, which binds the created ClusterRole psp-role to the created SA psp-sa.
Hint:
Also, Check the Configuration is working or not by trying to Mount a Secret in the pod maifest, it should get failed.
POD Manifest:
apiVersion: v1
kind: Pod
metadata:
name:
spec:
containers:
- name:
image:
volumeMounts:
- name:
mountPath:
volumes:
- name:
secret:
secretname:

Answer:

Explanation:
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
name: restricted
annotations:
seccomp.security.alpha.kubernetes.io/allowedProfileNames: 'docker/default,runtime/default'
apparmor.security.beta.kubernetes.io/allowedProfileNames: 'runtime/default'
seccomp.security.alpha.kubernetes.io/defaultProfileName: 'runtime/default'
apparmor.security.beta.kubernetes.io/defaultProfileName: 'runtime/default' spec:
privileged: false
Required to prevent escalations to root.
allowPrivilegeEscalation: false
This is redundant with non-root + disallow privilege escalation,
but we can provide it for defense in depth.
requiredDropCapabilities:

- ALL
Allow core volume types.
volumes:
- 'configMap'
- 'emptyDir'
- 'projected'
- 'secret'
- 'downwardAPI'
Assume that persistentVolumes set up by the cluster admin are safe to use.
- 'persistentVolumeClaim'
hostNetwork: false
hostIPC: false
hostPID: false
runAsUser:
Require the container to run without root privileges.
rule: 'MustRunAsNonRoot'
seLinux:
This policy assumes the nodes are using AppArmor rather than SELinux.
rule: 'RunAsAny'
supplementalGroups:
rule: 'MustRunAs'
ranges:
Forbid adding the root group.
- min: 1
max: 65535
fsGroup:
rule: 'MustRunAs'
ranges:
Forbid adding the root group.
- min: 1
max: 65535
readOnlyRootFilesystem: false

NEW QUESTION # 14
Context
AppArmor is enabled on the cluster's worker node. An AppArmor profile is prepared, but not enforced yet.
Task
On the cluster's worker node, enforce the prepared AppArmor profile located at /etc/apparmor.d/nginx_apparmor.
Edit the prepared manifest file located at /home/candidate/KSSH00401/nginx-pod.yaml to apply the AppArmor profile.
Finally, apply the manifest file and create the Pod specified in it.

Answer:

Explanation:

NEW QUESTION # 15
You are working on a Kubernetes cluster that has a deployment named 'web-app'. The deployment is currently running on a single node. You need to implement a pod disruption budget (PDB) for this deployment to ensure that at least 2 out of 3 pods are always available during a rolling update. How would you implement a pod disruption budget (PD8) to achieve this, and what commands would you use to ensure that at least 2 out of 3 pods are always available during a rolling update.

Answer:

Explanation:
Solution (Step by Step) :
1. Create a Pod Disruption Budget (PDB):
- Create a YAML file named 'web-app-pdb.yamr with the following content:

- Apply the PDB using 'kubectl apply -f web-app-pdb.yaml' 2. Verify the PDB Creation: - Use the command 'kubectl get pdb' to list all existing PDBs- - Check that the 'web-app-pdb' is listed With the desired configuration. 3. Initiate a Rolling Update: - Perform a rolling update for the 'web-app' deployment using the command 'kubectl rollout restart deployment web-apps 4. Monitor the IJpdate Process: - Use the command 'kuoectl get pods -l app=web-app' to monitor the status of the pods during the rolling update. - Ensure that at least two pods are always running, even during pod termination and replacement. 5. Check for PDB Enforcement: - If the rolling update tries to disrupt more than one pod, the PDB should prevent the update from proceeding. - You'll see an error message indicating that the disruption budget is being enforced.

NEW QUESTION # 16
SIMULATION
Cluster: dev
Master node: master1
Worker node: worker1
You can switch the cluster/configuration context using the following command:
[desk@cli] $ kubectl config use-context dev
Task:
Retrieve the content of the existing secret named adam in the safe namespace.
Store the username field in a file names /home/cert-masters/username.txt, and the password field in a file named /home/cert-masters/password.txt.
1. You must create both files; they don't exist yet.
2. Do not use/modify the created files in the following steps, create new temporary files if needed.
Create a new secret names newsecret in the safe namespace, with the following content:
Username: dbadmin
Password: moresecurepas
Finally, create a new Pod that has access to the secret newsecret via a volume:
Namespace: safe
Pod name: mysecret-pod
Container name: db-container
Image: redis
Volume name: secret-vol
Mount path: /etc/mysecret

Answer:

Explanation:
See the Explanation below
Explanation:

NEW QUESTION # 17
SIMULATION
Fix all issues via configuration and restart the affected components to ensure the new setting takes effect.
Fix all of the following violations that were found against the API server:- a. Ensure that the RotateKubeletServerCertificate argument is set to true.
b. Ensure that the admission control plugin PodSecurityPolicy is set.
c. Ensure that the --kubelet-certificate-authority argument is set as appropriate.
Fix all of the following violations that were found against the Kubelet:- a. Ensure the --anonymous-auth argument is set to false.
b. Ensure that the --authorization-mode argument is set to Webhook.
Fix all of the following violations that were found against the ETCD:-
a. Ensure that the --auto-tls argument is not set to true
b. Ensure that the --peer-auto-tls argument is not set to true
Hint: Take the use of Tool Kube-Bench

Answer:

Explanation:
Fix all of the following violations that were found against the API server:- a. Ensure that the RotateKubeletServerCertificate argument is set to true.
apiVersion: v1

kind: Pod
metadata:
creationTimestamp: null
labels:
component: kubelet
tier: control-plane
name: kubelet
namespace: kube-system
spec:
containers:
- command:
- kube-controller-manager
+ - --feature-gates=RotateKubeletServerCertificate=true
image: gcr.io/google_containers/kubelet-amd64:v1.6.0
livenessProbe:
failureThreshold: 8
httpGet:
host: 127.0.0.1
path: /healthz
port: 6443
scheme: HTTPS
initialDelaySeconds: 15
timeoutSeconds: 15
name: kubelet
resources:
requests:
cpu: 250m
volumeMounts:
- mountPath: /etc/kubernetes/
name: k8s
readOnly: true
- mountPath: /etc/ssl/certs
name: certs
- mountPath: /etc/pki
name: pki
hostNetwork: true
volumes:
- hostPath:
path: /etc/kubernetes
name: k8s
- hostPath:
path: /etc/ssl/certs
name: certs
- hostPath:
path: /etc/pki
name: pki
b. Ensure that the admission control plugin PodSecurityPolicy is set.
audit: "/bin/ps -ef | grep $apiserverbin | grep -v grep"
tests:
test_items:
- flag: "--enable-admission-plugins"
compare:
op: has
value: "PodSecurityPolicy"
set: true
remediation: |
Follow the documentation and create Pod Security Policy objects as per your environment.
Then, edit the API server pod specification file $apiserverconf
on the master node and set the --enable-admission-plugins parameter to a value that includes PodSecurityPolicy :
--enable-admission-plugins=...,PodSecurityPolicy,...
Then restart the API Server.

scored: true
c. Ensure that the --kubelet-certificate-authority argument is set as appropriate.
audit: "/bin/ps -ef | grep $apiserverbin | grep -v grep"
tests:
test_items:
- flag: "--kubelet-certificate-authority"
set: true
remediation: |
Follow the Kubernetes documentation and setup the TLS connection between the apiserver and kubelets. Then, edit the API server pod specification file
$apiserverconf on the master node and set the --kubelet-certificate-authority parameter to the path to the cert file for the certificate authority.
--kubelet-certificate-authority=<ca-string>
scored: true
Fix all of the following violations that were found against the ETCD:-
a. Ensure that the --auto-tls argument is not set to true
Edit the etcd pod specification file $etcdconf on the master node and either remove the --auto-tls parameter or set it to false. --auto-tls=false b. Ensure that the --peer-auto-tls argument is not set to true Edit the etcd pod specification file $etcdconf on the master node and either remove the --peer-auto-tls parameter or set it to false. --peer-auto-tls=false

NEW QUESTION # 18
.....