Free PDF Quiz Accurate Linux Foundation - CKS Training Kit

P.S. Free & New CKS dumps are available on Google Drive shared by Prep4sureExam: https://drive.google.com/open?id=1CV3tzE_PIjy4FaWpGELX1AwfnHPOG7YF

Our CKS practice questions are carfully compiled by our professional experts to be sold all over the world. So the content should be easy to be understood. The difficult questions of the CKS exam materials will have vivid explanations. So you will have a better understanding after you carefully see the explanations. At the same time, our CKS Real Exam just needs to cost you a few spare time. After about twenty to thirty hours' practice, you can completely master all knowledge.

The CKS Exam is an important certification for anyone who works with Kubernetes and wants to demonstrate their expertise in securing these environments. By passing the exam, individuals can prove to potential employers that they have the skills and knowledge needed to secure Kubernetes clusters and protect the applications that run on them, making them a valuable asset to any organization that uses Kubernetes for their container orchestration and deployment needs.

>> CKS Training Kit <<

Latest Linux Foundation CKS Exam Questions, New CKS Test Pdf

We cannot overlook the importance of efficiency because we live in a society emphasize on it. So to get our latest CKS exam torrent, just enter the purchasing website, and select your favorite version with convenient payment and you can download our latest CKS exam torrent immediately within 5 minutes. This way you can avoid the problems in waiting for arrival of products and you can learn about the knowledge of CKS Quiz guides in a short time. Latest CKS exam torrent can vividly embody the spirits and effort

we have put into them. And the power of our CKS test prep permit you to apprehend the essence of the exam. All elites in this area vindicate the accuracy and efficiency of our CKS quiz guides.

Linux Foundation Certified Kubernetes Security Specialist (CKS) Sample Questions (Q31-Q36):

NEW QUESTION # 31
You are running a highly sensitive application in your Kubernetes cluster, which stores personal identifiable information (PII) dat a. You suspect that a malicious actor might have injected a malicious container image into your cluster and is now attempting to exfiltrate this data. You need to implement a solution to detect and prevent any suspicious data exfiltration attempts from within your cluster.

Answer:

Explanation:
Solution (Step by Step):
1. Enable Container Security Policies (CSP) with Admission Control:
- Configure a CSP policy using the 'PodSecurityPolicy' or the newer 'PodSecurity' object.
- Restrict network egress for containers running your sensitive application to only allow communication to approved external services and destinations.
- Define rules within the CSP policy tnat disallow any container from accessing privileged ports or using privileged capabilities. This will limit the
attackers ability to establish unauthorized connections or manipulate system resources.
- Example CSP policy With 'P0dSecurity'
2. Implement Network Policies: - Configure network policies to restrict outbound network traffic from pods running tne sensitive application. - Allow only specific ports and destinations required tor the application's functionality. - This step helps prevent any unauthorized connections from the compromised container to external networks. - Example Network Policy:
3. Deploy Intrusion Detection Systems (IDS) in the Cluster: - Deploy an IDS solution like Falco or Sysdig within your cluster. - Configure Falco to monitor for suspicious activities like file system modifications, network connections, or process executions that might indicate data exfiltration attempts. - Falco can trigger alerts and block malicious activities based on the defined rules. - Example Falco rule:
4. Utilize Runtime Security Tools: - Deploy a runtime security tool like Aqua Security, Twistlock, or Snyk. - These tools monitor running containers for suspicious behaviors and vulnerabilities. - They can enforce security policies, detect anomalies, and alen you about potential data breaches. - This helps you quickly identify compromised containers and take appropriate actions. 5. Implement Data Encryption and Access Control: - Encrypt the PII data stored in your Kubernetes cluster at rest and in transit - Utilize tools like Vault or KMS to manage and secure encryption keys. - Implement access control measures to limit access to sensitive data to authorized users and applications. - This minimizes the impact of a data breach even if the malicious container gains access to the data. By combining these security measures, you can significantly reduce the risk of data exfiltration and enhance the security posture of your sensitive application running in the Kubernetes cluster.

NEW QUESTION # 32
SIMULATION
Create a network policy named allow-np, that allows pod in the namespace staging to connect to port 80 of other pods in the same namespace.
Ensure that Network Policy:-
1. Does not allow access to pod not listening on port 80.
2. Does not allow access from Pods, not in namespace staging.

Answer:

Explanation:
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: network-policy
spec:
podSelector: {} #selects all the pods in the namespace deployed
policyTypes:
- Ingress

ingress:
- ports: #in input traffic allowed only through 80 port only
- protocol: TCP
port: 80

NEW QUESTION # 33
You are tasked with securing a Kubernetes cluster running kube-dns. You need to enforce the CIS Kubernetes Benchmark recommendations for kube-dns. One of the key recommendations is to disable the '-bind-address-0.0.0.0' parameter from the kube-dns deployment. This parameter allows kube-dns to listen on all network interfaces, potentially exposing the DNS service to unwanted access. How would you achieve this using a ConfigMap?

Answer:

Explanation:
Solution (Step by Step) :
1. Create a ConfigMap: Create a ConfigMap named 'kube-dns-config' containing the updated configuration for kube-dns. This ConfigMap will replace the default kube-dns configuration.
2 Apply the ConfigMap: Apply the ConfigMap to the cluster using 'kubectl apply -f kube-dns-config-yaml'. This will create the ConfigMap and update the kube-dns deployment. 3. Verify the Deployment: Verify that the kube-dns deployment has been updated with the new configuration. Use 'kubectl get deployment kube-dns -o yaml' to see the deployment configuration, and Check tor the "-bind-address=127.O.O.1 ' parameter in the container's command. 4. Restart the kube-dns Pods: Restart the kube-dns pods to ensure the changes take effect This can be done using the 'kubectl rollout restart deployment kube-dns' command. This change will ensure that kube-dns is only listening on the localhost interface (127.0.0.1), mitigating the risk of unauthorized access.

NEW QUESTION # 34
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 # 35
You can switch the cluster/configuration context using the following command:
[desk@cli] $ kubectl config use-context test-account
Task: Enable audit logs in the cluster.
To do so, enable the log backend, and ensure that:
1. logs are stored at /var/log/Kubernetes/logs.txt
2. log files are retained for 5 days
3. at maximum, a number of 10 old audit log files are retained
A basic policy is provided at /etc/Kubernetes/logpolicy/audit-policy.yaml. It only specifies what not to log.
Note: The base policy is located on the cluster's master node.
Edit and extend the basic policy to log:
1. Nodes changes at RequestResponse level

2. The request body of persistentvolumes changes in the namespace frontend
3. ConfigMap and Secret changes in all namespaces at the Metadata level Also, add a catch-all rule to log all other requests at the Metadata level Note: Don't forget to apply the modified policy.

Answer:

Explanation:
$ vim /etc/kubernetes/log-policy/audit-policy.yaml
- level: RequestResponse
userGroups: ["system:nodes"]
- level: Request
resources:
- group: "" # core API group
resources: ["persistentvolumes"]
namespaces: ["frontend"]
- level: Metadata
resources:
- group: ""
resources: ["configmaps", "secrets"]
- level: Metadata
$ vim /etc/kubernetes/manifests/kube-apiserver.yaml
Add these
- --audit-policy-file=/etc/kubernetes/log-policy/audit-policy.yaml
- --audit-log-path=/var/log/kubernetes/logs.txt
- --audit-log-maxage=5
- --audit-log-maxbackup=10
Explanation
[desk@cli] $ ssh master1
[master1@cli] $ vim /etc/kubernetes/log-policy/audit-policy.yaml
apiVersion: audit.k8s.io/v1 # This is required.
kind: Policy
Don't generate audit events for all requests in RequestReceived stage.
omitStages:
- "RequestReceived"
rules:
Don't log watch requests by the "system:kube-proxy" on endpoints or services
- level: None
users: ["system:kube-proxy"]
verbs: ["watch"]
resources:
- group: "" # core API group
resources: ["endpoints", "services"]
Don't log authenticated requests to certain non-resource URL paths.
- level: None
userGroups: ["system:authenticated"]
nonResourceURLs:
- "/api*" # Wildcard matching.
- "/version"
Add your changes below
- level: RequestResponse
userGroups: ["system:nodes"] # Block for nodes
- level: Request
resources:
- group: "" # core API group
resources: ["persistentvolumes"] # Block for persistentvolumes
namespaces: ["frontend"] # Block for persistentvolumes of frontend ns
- level: Metadata
resources:
- group: "" # core API group
resources: ["configmaps", "secrets"] # Block for configmaps & secrets
- level: Metadata # Block for everything else

[master1@cli] $ vim /etc/kubernetes/manifests/kube-apiserver.yaml
apiVersion: v1
kind: Pod
metadata:
annotations:
kubeadm.kubernetes.io/kube-apiserver.advertise-address.endpoint: 10.0.0.5:6443 labels:
component: kube-apiserver
tier: control-plane
name: kube-apiserver
namespace: kube-system
spec:
containers:
- command:
- kube-apiserver
- --advertise-address=10.0.0.5
- --allow-privileged=true
- --authorization-mode=Node,RBAC
- --audit-policy-file=/etc/kubernetes/log-policy/audit-policy.yaml #Add this
- --audit-log-path=/var/log/kubernetes/logs.txt #Add this
- --audit-log-maxage=5 #Add this
- --audit-log-maxbackup=10 #Add this
...
output truncated
Note: log volume & policy volume is already mounted in vim /etc/kubernetes/manifests/kube-apiserver.yaml so no need to mount it. Reference: https://kubernetes.io/docs/tasks/debug-application-cluster/audit/ Note: log volume & policy volume is already mounted in vim /etc/kubernetes/manifests/kube-apiserver.yaml so no need to mount it. Reference: https://kubernetes.io/docs/tasks/debug-application-cluster/audit/

NEW QUESTION # 36
.....

CKS □ to download exam materials for free □CKS Exam Simulator Fee

- myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, myportal.utt.edu.tt, acgwg.com, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, dl.instructure.com, zenwriting.net, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, Disposable vapes

BTW, DOWNLOAD part of Prep4sureExam CKS dumps from Cloud Storage: https://drive.google.com/open?id=1CV3tzE_PIjy4FaWpGELX1AwfnHPOG7YF