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.pass2lead.com/cks.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Linux Foundation Official Exam Center

QUESTION 1

Enable audit logs in the cluster, To Do so, enable the log backend, and ensure that

1.

logs are stored at /var/log/kubernetes/kubernetes-logs.txt.

2.

Log files are retained for 5 days.

3.

at maximum, a number of 10 old audit logs files are retained. Edit and extend the basic policy to log:

1.

Cronjobs changes at RequestResponse

2.

Log the request body of deployments changes in the namespace kube-system.

3.

Log all other resources in core and extensions at the Request level.

4.

Don\\'t log watch requests by the "system:kube-proxy" on endpoints or

A. See explanation below.

B. PlaceHolder

Correct Answer: A

QUESTION 2

Create a User named john, create the CSR Request, fetch the certificate of the user after approving it.

Create a Role name john-role to list secrets, pods in namespace john

Finally, Create a RoleBinding named john-role-binding to attach the newly created role john-role to the user john in the namespace john.

To Verify: Use the kubectl auth CLI command to verify the permissions.

A. See the below.

B. PlaceHolder

Correct Answer: A

se kubectl to create a CSR and approve it.

Get the list of CSRs:

kubectl get csr

Approve the CSR:

kubectl certificate approve myuser

Get the certificateRetrieve the certificate from the CSR:

kubectl get csr/myuser -o yaml

here are the role and role-binding to give john permission to create NEW_CRD resource:

kubectl apply -f roleBindingJohn.yaml --as=john

rolebinding.rbac.authorization.k8s.io/john_external-rosource-rb created

kind: RoleBinding

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: john_crd

namespace: development-john

subjects:

-kind: User name: john apiGroup: rbac.authorization.k8s.io roleRef: kind: ClusterRole name: crd-creation

kind: ClusterRole apiVersion: rbac.authorization.k8s.io/v1 metadata: name: crd-creation rules:

-apiGroups: ["kubernetes-client.io/v1"] resources: ["NEW_CRD"] verbs: ["create, list, get"]

QUESTION 3

Create a new ServiceAccount named backend-sa in the existing namespace default, which has the capability to list the pods inside the namespace default.

Create a new Pod named backend-pod in the namespace default, mount the newly created sa backend-sa to the pod, and Verify that the pod is able to list pods.

Ensure that the Pod is running.

A. See the below:

B. PlaceHolder

Correct Answer: A

A service account provides an identity for processes that run in a Pod.

When you (a human) access the cluster (for example, using kubectl), you are authenticated by the apiserver as a particular User Account (currently this is usually admin, unless your cluster administrator has customized your cluster). Processes in containers inside pods can also contact the apiserver. When they do, they are authenticated as a particular Service Account (for example, default).

When you create a pod, if you do not specify a service account, it is automatically assigned the default service account in the same namespace. If you get the raw json or yaml for a pod you have created (for example, kubectl get pods/ -o yaml), you can see the spec.serviceAccountName field has been automatically set. You can access the API from inside a pod using automatically mounted service account credentials, as described in Accessing the Cluster. The API permissions of the service account depend on the authorization plugin and policy in use. In version 1.6+, you can opt out of automounting API credentials for a service account by setting automountServiceAccountToken: false on the service account:

apiVersion: v1 kind: ServiceAccount metadata: name: build-robot automountServiceAccountToken: false

In version 1.6+, you can also opt out of automounting API credentials for a particular pod: apiVersion: v1 kind: Pod metadata: name: my-pod spec: serviceAccountName: build-robot automountServiceAccountToken: false

The pod spec takes precedence over the service account if both specify a automountServiceAccountToken value.

QUESTION 4

You must complete this task on the following cluster/nodes: Cluster: immutable-cluster

Master node: master1

Worker node: worker1

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

[desk@cli] $ kubectl config use-context immutable-cluster

Context: It is best practice to design containers to be stateless and immutable.

Task:

Inspect Pods running in namespace prod and delete any Pod that is either not stateless or not immutable.

Use the following strict interpretation of stateless and immutable:

1.

Pods being able to store data inside containers must be treated as not stateless.

Note: You don\\'t have to worry whether data is actually stored inside containers or not already.

2.

Pods being configured to be privileged in any way must be treated as potentially not stateless or not immutable.

A. See the explanation below

B. PlaceHolder

Correct Answer: A

Explanation/Reference:

```
candidate@cli:~$ kubectl config use-context KSRS00501
Switched to context "KSRS00501".
candidate@cli:~$ kubectl get pod -n testing
NAME       READY   STATUS    RESTARTS   AGE
app        1/1     Running   0          6h31m
frontend   1/1     Running   0          6h32m
smtp       1/1     Running   0          6h31m
candidate@cli:~$ kubectl get pod/app -n testing  -o yaml
   - lastProbeTime: null
     lastTransitionTime: "2022-05-20T08:40:35Z"
     status: "True"
     type: PodScheduled
   containerStatuses:
   - containerID: docker://11143682c400984c9faf3dff1e056d4b00a7eb1de007fe1834be0a84fa146e18
     image: nginx:latest
     imageID: docker-pullable://nginx@sha256:2d17cc4981bf1e22a87ef3b3dd20fbb72c3868738e3f3076
62eb40e2630d4320
     lastState: {}
     name: app-container
     ready: true
     restartCount: 0
     started: true
     state:
       running:
         startedAt: "2022-05-20T08:40:37Z"
   hostIP: 10.240.86.141
   phase: Running
   podIP: 10.10.1.3
   podIPs:
   - ip: 10.10.1.3
   qosClass: BestEffort
   startTime: "2022-05-20T08:40:35Z"
candidate@cli:~$ kubectl get pod/app -n testing -o yaml | grep -E 'privileged|ReadOnlyFileSy
stem'
      privileged: true
candidate@cli:~$ kubectl get pod/frontend -n testing -o yaml | grep -E 'privileged|ReadOnlyF
ileSystem'
      privileged: false
```

```
candidate@cli:~$ kubectl get pod/smtp -n testing -o yaml | grep -E 'privileged|ReadOnlyFileS
ystem'
      privileged: true
candidate@cli:~$ kubectl get pod -n testing -o yaml | grep -i ReadOnly
        readOnlyRootFilesystem: false
        readOnly: true
        readOnlyRootFilesystem: true
        readOnly: true
        readOnlyRootFilesystem: false
        readOnly: true
candidate@cli:~$ kubectl get pod/smtp -n testing -o yaml | grep -E 'privileged|readOnlyRootF
ileSystem'
      privileged: true
candidate@cli:~$ kubectl get pod/app -n testing -o yaml | grep -E 'privileged|readOnlyRootFi
leSystem'
      privileged: true
candidate@cli:~$ kubectl get pod/frontend -n testing -o yaml | grep -E 'privileged|readOnlyR
ootFileSystem'
      privileged: false
candidate@cli:~$ kubectl get pod/frontend -n testing -o yaml | grep -E 'privileged|readOnlyR
ootFilesystem'
      privileged: true
      readOnlyRootFilesystem: false
candidate@cli:~$ kubectl delete pod/app -n testing
pod "app" deleted
candidate@cli:~$ kubectl get pod/smtp -n testing -o yaml | grep -E 'privileged|readOnlyRootF
ilesystem'
      privileged: true
      readOnlyRootFilesystem: false
candidate@cli:~$ kubectl delete pod/smtp -n testing
pod "smtp" deleted
```

QUESTION 5

You must complete this task on the following cluster/nodes:

Cluster: trace Master node: master Worker node: worker1

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

[desk@cli] $ kubectl config use-context trace

Given: You may use Sysdig or Falco documentation.

Task:

Use detection tools to detect anomalies like processes spawning and executing something weird frequently in the single container belonging to Pod tomcat.

Two tools are available to use:

1.

falco

2.

sysdig

Tools are pre-installed on the worker1 node only.

Analyse the container\\'s behaviour for at least 40 seconds, using filters that detect newly spawning and executing processes.

Store an incident file at /home/cert_masters/report, in the following format:

[timestamp],[uid],[processName]

Note: Make sure to store incident file on the cluster\\'s worker node, don\\'t move it to master node.

A. See the explanation below

B. PlaceHolder

Correct Answer: A

$vim /etc/falco/falco_rules.local.yaml uk.co.certification.simulator.questionpool.PList@120e24d0 $kill -1 Explanation[desk@cli] $ ssh node01[node01@cli] $ vim /etc/falco/falco_rules.yamlsearch for Container Drift Detected and paste in falco_rules.local.yaml[node01@cli] $ vim /etc/falco/falco_rules.local.yaml

-rule: Container Drift Detected (open+create) desc: New executable created in a container due to open+create condition: > 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 Practice Test CKS Study Guide CKS Exam Questions