Exam Questions KCSA

Kubernetes and Cloud Native Security Associate (KCSA)

https://www.2passeasy.com/dumps/KCSA/

NEW QUESTION 1
What information is stored in etcd?

A. Etcd manages the configuration data, state data, and metadata for Kubernetes.
B. Application logs and monitoring data for auditing and troubleshooting purposes.
C. Sensitive user data such as usernames and passwords.
D. Pod data contained in Persistent Volume Claims (e.
E. hostPath).

Answer: A

Explanation:

- etcdis Kubernetes??key-value storeforcluster state.
- Stores: ConfigMaps, Secrets, Pod definitions, Deployments, RBAC policies, and metadata.
- Exact extract (Kubernetes Docs – etcd):
- ??etcd is a consistent and highly-available key-value store used as Kubernetes?? backing store for all cluster data.??
- Clarifications:
- B: Logs/metrics are handled by logging/monitoring solutions, not etcd.
- C: Secrets may be stored here but encoded in base64, not specifically "usernames/passwords" as primary use.
- D: Persistent Volumes are external storage, not stored in etcd.

References:
Kubernetes Docs — etcd: https://kubernetes.io/docs/concepts/overview/components/#etcd

NEW QUESTION 2
Which information does a user need to verify a signed container image?

A. The image's SHA-256 hash and the private key of the signing authority.
B. The image's digital signature and the private key of the signing authority.
C. The image's SHA-256 hash and the public key of the signing authority.
D. The image's digital signature and the public key of the signing authority.

Answer: D

Explanation:

- Container image signing (e.g., withcosign, Notary v2) uses asymmetric cryptography.
- Verification process:
- Retrieve theimage??s digital signature.
- Validate the signature with thepublic keyof the signer.
- Exact extract (Sigstore Cosign Docs):
- ??Verification of an image requires the signature and the signer??s public key. The signature proves authenticity and integrity.??
- Why others are wrong:
- A & B: The private key is only used by the signer, never shared.
- C: The hash alone cannot prove authenticity without the digital signature.

References:
Sigstore Cosign Docs: https://docs.sigstore.dev/cosign/overview

NEW QUESTION 3
A cluster is failing to pull more recent versions of images from k8s.gcr.io. Why may this be?

A. There is a network connectivity issue between the cluster and k8s.gcr.io.
B. There is a bug in the container runtime or the image pull process.
C. The authentication credentials for accessing k8s.gcr.io are incorrectly scoped.
D. The container image registry k8s.gcr.io has been deprecated.

Answer: D

Explanation:

- k8s.gcr.iowas the historic Kubernetes image registry.
- It has beendeprecatedand replaced withregistry.k8s.io.

- Exact extract (Kubernetes Blog):
- ??The k8s.gcr.io image registry will be frozen from April 3, 2023 and fully deprecated. All Kubernetes project images are now served from registry.k8s.io.??
- Pulling newer versions from k8s.gcr.io fails because the registry no longer receives updates.

References:
Kubernetes Blog — Image Registry Update: https://kubernetes.io/blog/2023/02/06/k8s-gcr-io-freeze- announcement/

NEW QUESTION 4
In a Kubernetes environment, what kind of Admission Controller can modify resource manifests when applied to the Kubernetes API to fix misconfigurations automatically?

A. ValidatingAdmissionController
B. PodSecurityPolicy
C. MutatingAdmissionController
D. ResourceQuota

Answer: C

Explanation:
- Kubernetes Admission Controllers can eithervalidateormutateincoming requests.
- MutatingAdmissionWebhook (Mutating Admission Controller):
- Canmodify or mutate resource manifestsbefore they are persisted in etcd.
- Used for automatic injection of sidecars (e.g., Istio Envoy proxy), setting default values, or fixing misconfigurations.
- ValidatingAdmissionWebhook (Validating Admission Controller):only allows/denies but doesnot change requests.
- PodSecurityPolicy:deprecated; cannot mutate requests.
- ResourceQuota:enforces resource usage, but does not mutate manifests.

Exact Extract:
- ??Mutating admission webhooks are invoked first, and can modify objects to enforce defaults.

Validating admission webhooks are invoked second, and can reject requests to enforce invariants.??
References:
Kubernetes Docs — Admission Controllers: https://kubernetes.io/docs/reference/access-authn-authz
/admission-controllers/
Kubernetes Docs — Admission Webhooks: https://kubernetes.io/docs/reference/access-authn-authz
/extensible-admission-controllers/

NEW QUESTION 5
In a Kubernetes cluster, what are the security risks associated with using ConfigMaps for storing secrets?

A. Storing secrets in ConfigMaps does not allow for fine-grained access control via RBAC.
B. Storing secrets in ConfigMaps can expose sensitive information as they are stored in plaintext and can be accessed by unauthorized users.
C. Using ConfigMaps for storing secrets might make applications incompatible with the Kubernetes cluster.
D. ConfigMaps store sensitive information in etcd encoded in base64 format automatically, which does not ensure confidentiality of data.

Answer: B

Explanation:
- ConfigMaps are explicitly not for confidential data.
- Exact extract (ConfigMap concept):"A ConfigMap is an API object used to store non-confidential data in key-value pairs."
- Exact extract (ConfigMap concept):"ConfigMaps are not intended to hold confidential data. Use a Secret for confidential data."
- Why this is risky:data placed into a ConfigMap is stored as regular (plaintext) string values in the API and etcd (unless you deliberately use binaryData for base64 content you supply). That means if someone has read access to the namespace or to etcd/APIServer storage, they can view the values.
- Secrets vs ConfigMaps (to clarify distractor D):
- Exact extract (Secret concept):"By default, secret data is stored as unencrypted base64-encoded strings.You canenable encryption at restto protect Secrets stored in etcd."
- This base64 behavior applies toSecrets, not to ConfigMap data. Thus optionDis incorrect for ConfigMaps.
- About RBAC (to clarify distractor A):Kubernetesdoessupport fine-grained RBAC forbothConfigMaps and Secrets; the issue isn't lack of RBAC but that ConfigMaps arenotdesigned for confidential material.
- About compatibility (to clarify distractor C):Using ConfigMaps for secrets doesn't make apps "incompatible"; it's simplyinsecureand against guidance.

[References:, Kubernetes Docs —ConfigMaps: https://kubernetes.io/docs/concepts/configuration/configmap/, Kubernetes Docs —Secrets: https://kubernetes.io/docs/concepts/configuration/secret/, Kubernetes Docs —Encrypting Secret Data at Rest: https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data/, Note: The citations above are from the official Kubernetes documentation and reflect the stated guidance that ConfigMaps are fornon-confidentialdata, while Secrets (with encryption at rest enabled) are forconfidentialdata, and that the 4C's map todefense in depth.,]

NEW QUESTION 6
What is the difference between gVisor and Firecracker?

A. gVisor is a user-space kernel that provides isolation and security for container
B. At the same time, Firecracker is a lightweight virtualization technology for creating and managing secure, multi-tenant container and function-as-a-service (FaaS) workloads.
C. gVisor is a lightweight virtualization technology for creating and managing secure, multi-tenant container and function-as-a-service (FaaS) workload
D. At the same time, Firecracker is a user-space kernel that provides isolation and security for containers.
E. gVisor and Firecracker are both container runtimes that can be used interchangeably.
F. gVisor and Firecracker are two names for the same technology, which provides isolation and security for containers.

Answer: A

Explanation:
gVisor:
Google-developed, implemented as auser-space kernelthat intercepts and emulates syscalls made by containers.
Providesstrong isolationwithout requiring a full VM.
Official docs: "gVisor is a user-space kernel, written in Go, that implements a substantial portion of the Linux system call interface."
Source: https://gvisor.dev/docs/
Firecracker:
AWS-developed,lightweight virtualization technologybuilt on KVM, used in AWS Lambda and Fargate.
Optimized for running secure, multi-tenant microVMs (MicroVMs) for containers and FaaS.
Official docs: "Firecracker is an open-source virtualization technology that is purpose-built for creating and managing secure, multi-tenant container and function-based services."
Source: https://firecracker-microvm.github.io/
Key difference:gVisor ?? syscall interception in userspace kernel (container isolation). Firecracker ?? lightweight virtualization with microVMs (multi-tenant security).
Therefore, optionAis correct.
[References:, gVisor Docs: https://gvisor.dev/docs/, Firecracker Docs: https://firecracker-microvm.github.io/,]

NEW QUESTION 7
Which technology can be used to apply security policy for internal cluster traffic at the application layer of the network?

A. Network Policy
B. Ingress Controller
C. Container Runtime
D. Service Mesh

Answer: D

Explanation:
Service Mesh (e.g., Istio, Linkerd, Consul):operates atLayer 7 (application layer), enforcing policies like mTLS, authorization, and routing between services.
NetworkPolicy:works atLayer 3/4 (IP/port), not Layer 7.
Ingress Controller:handles external traffic ingress, not internal service-to-service traffic.
Container Runtime:responsible for running containers, not enforcing application-layer security.
Exact extract (Istio docs):
"Istio provides security by enforcing authentication, authorization, and encryption of service-to-service communication."
[References:, Kubernetes Docs — Network Policies: https://kubernetes.io/docs/concepts/services-networking/network-policies/, Istio Security Docs: https://istio.io/latest/docs/concepts/security/,]

NEW QUESTION 8
In the event that kube-proxy is in a CrashLoopBackOff state, what impact does it have on the Pods running on the same worker node?

A. The Pods cannot communicate with other Pods in the cluster.
B. The Pod cannot mount persistent volumes through CSI drivers.
C. The Pod's security context restrictions cannot be enforced.
D. The Pod's resource utilization increases significantly.

Answer: A

Explanation:
kube-proxy:manages cluster network routing rules (via iptables or IPVS). It enables Pods to communicate with Services and Pods across nodes.
If kube-proxy fails (CrashLoopBackOff), service IP routing and cluster-wide pod-to-pod networking breaks. Local Pod-to-Pod communication within the same node may still work, butcross-node communication fails.
Exact extract (Kubernetes Docs – kube-proxy):
"kube-proxy maintains network rules on nodes. These rules allow network communication to Pods from network sessions inside or outside of the cluster."
[References:, Kubernetes Docs — kube-proxy: https://kubernetes.io/docs/reference/command-line-tools-reference/kube-proxy/,]

NEW QUESTION 9
When using a cloud provider's managed Kubernetes service, who is responsible for maintaining the etcd cluster?

A. Kubernetes administrator
B. Namespace administrator
C. Cloud provider
D. Application developer

Answer: C

Explanation:
Inmanaged Kubernetes services(EKS, GKE, AKS), the control plane is operated by thecloud provider.
This includesetcd, API server, controller manager, scheduler.
Users manageworker nodes(in some models) and workloads, but not the control plane.
Exact extract (GKE Docs):
"The control plane, including the API server and etcd database, is managed and maintained by Google."
Similarly forEKSandAKS, etcd is fully managed by the provider.
[References:, GKE Architecture: https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-architecture, EKS Architecture: https://docs.aws.amazon.com/eks/latest/userguide/eks-architecture.html, AKS Docs: https://learn.microsoft.com/en-us/azure/aks/concepts-clusters-workloads,]

NEW QUESTION 10
A Kubernetes cluster tenant can launch privileged Pods in contravention of therestricted Pod Security Standardmandated for cluster tenants and enforced by the built-inPodSecurity admission controller.
The tenant has full CRUD permissions on the namespace object and the namespaced resources. How did the tenant achieve this?

A. The scope of the tenant role means privilege escalation is impossible.
B. By tampering with the namespace labels.
C. By deleting the PodSecurity admission controller deployment running in their namespace.
D. By using higher-level access credentials obtained reading secrets from another namespace.

Answer: B

Explanation:
ThePodSecurity admission controllerenforces Pod Security Standards (Baseline, Restricted, Privileged)based on namespace labels.
If a tenant has full CRUD on the namespace object, they canmodify the namespace labelsto remove or weaken the restriction (e.g., setting pod-security.kubernetes.io/enforce=privileged).
This allows privileged Pods to be admitted despite the security policy.
Incorrect options:
(A) is false — namespace-level access allows tampering.
(C) is invalid — PodSecurity admission is not namespace-deployed, it??s a cluster-wide admission controller.
(D) is unrelated — Secrets from other namespaces wouldn??t directly bypass PodSecurity enforcement.
References:
Kubernetes Documentation – Pod Security Admission
CNCF Security Whitepaper – Admission control and namespace-level policy enforcement weaknesses.

NEW QUESTION 10
What mechanism can I use to block unsigned images from running in my cluster?

A. Enabling Admission Controllers to validate image signatures.
B. Using PodSecurityPolicy (PSP) to enforce image signing and validation.
C. Using Pod Security Standards (PSS) to enforce validation of signatures.
D. Configuring Container Runtime Interface (CRI) to enforce image signing and validation.

Answer: A

Explanation:
KubernetesAdmission Controllers(particularlyValidatingAdmissionWebhooks) can be used to enforce policies that validate image signatures.
This is commonly implemented withtools like Sigstore/cosign, Kyverno, or OPA Gatekeeper.
PodSecurityPolicy (PSP):deprecated and never supported image signature validation.
Pod Security Standards (PSS):only apply to pod security fields (privilege, users, host access), not image signatures.
CRI:while runtimes (containerd, CRI-O) may integrate with signature verification tools, enforcement in Kubernetes is generally done viaAdmission Controllersat the API layer.
Exact extract (Admission Controllers docs):
??Admission webhooks can be used to enforce custom policies on the objects being admitted.?? (e.g., validating signatures).
References:
Kubernetes Docs — Admission Controllers: https://kubernetes.io/docs/reference/access-authn-authz/admission-controllers/
Sigstore Project (cosign): https://sigstore.dev/
Kyverno ImageVerify Policy: https://kyverno.io/policies/pod-security/require-image-verification/

NEW QUESTION 13
Which label should be added to the Namespace to block any privileged Pods from being created in that Namespace?

A. privileged: false
B. privileged: true
C. pod-security.kubernetes.io/enforce: baseline
D. pod.security.kubernetes.io/privileged: false

Answer: C

Explanation:
KubernetesPod Security Admission (PSA)enforcesPod Security Standardsby applying labels on Namespaces.

Exact extract (Kubernetes Docs – Pod Security Admission):
??You can label a namespace with pod-security.kubernetes.io/enforce: baseline to enforce the Baseline policy.??
Thebaselineprofile explicitly disallowsprivileged podsand other unsafe features.
Why others are wrong:
A & D: These labels do not exist in Kubernetes.
B: Setting privileged: true would allow privileged pods, not block them.
References:
Kubernetes Docs — Pod Security Admission: https://kubernetes.io/docs/concepts/security/pod-security- admission/
Kubernetes Docs — Pod Security Standards: https://kubernetes.io/docs/concepts/security/pod-security- standards/

NEW QUESTION 14
What is the reasoning behind considering the Cloud as the trusted computing base of a Kubernetes cluster?

A. The Cloud enforces security controls at the Kubernetes cluster level, so application developers can focus on applications only.
B. A Kubernetes cluster can only be trusted if the underlying Cloud provider is certified against international standards.
C. A vulnerability in the Cloud layer has a negligible impact on containers due to Linux isolation mechanisms.
D. A Kubernetes cluster can only be as secure as the security posture of its Cloud hosting.

Answer: D

Explanation:
The4C??s of Cloud Native Security(Cloud, Cluster, Container, Code) model starts withCloudas the base layer.
If the Cloud (infrastructure layer) is compromised, every higher layer (Cluster, Container, Code) inherits that compromise.
Exact extract (Kubernetes Security Overview):
??The 4C??s of Cloud Native security are Cloud, Clusters, Containers, and Code. You can think of the 4C??s as a layered approach. A Kubernetes cluster can only be as secure as the cloud infrastructure it is deployed on.??
This means the cloud is part of thetrusted computing baseof a Kubernetes cluster.
References:
Kubernetes Docs — Security Overview (4C??s): https://kubernetes.io/docs/concepts/security/overview/#the- 4cs-of-cloud-native-security

NEW QUESTION 15
Which of the following is a valid security risk caused by having no egress controls in a Kubernetes cluster?

A. Denial of Service
B. Data exfiltration
C. Increased attack surface
D. Unauthorized access to external resources

Answer: B

Explanation:
Egress NetworkPoliciesrestrict outbound traffic from Pods.
Without egress restrictions, a compromised Pod could exfiltrate sensitive data (secrets, logs, customer data) to an attacker-controlled server.
Exact extract (Kubernetes Docs – Network Policies):
"Egress rules control outbound connections from Pods. Without such restrictions, compromised workloads can connect freely to external endpoints."
Other options clarified:
A: DoS is more about flooding, not egress absence.
C: ??Increased attack surface?? is vague but not the main risk.
D: True in a sense, but the precise and most common risk isdata exfiltration.
[References:, Kubernetes Docs — Network Policies: https://kubernetes.io/docs/concepts/services-networking/network-policies/,]

NEW QUESTION 16
What is the purpose of an egress NetworkPolicy?

A. To control the incoming network traffic to a Kubernetes cluster.
B. To control the outbound network traffic from a Kubernetes cluster.
C. To secure the Kubernetes cluster against unauthorized access.
D. To control the outgoing network traffic from one or more Kubernetes Pods.

Answer: D

Explanation:
NetworkPolicycontrols network trafficat the Pod level.
Ingress rules:controlincomingconnections to Pods.
Egress rules:controloutgoingconnectionsfrom Pods.
Exact extract (Kubernetes Docs – Network Policies):
"An egress rule controls outgoing connections from Pods that match the policy."
Clarifying wrong answers:
A/B: Too broad (cluster-level); policies apply per Pod/Namespace.
C: Security against unauthorized access is broader than egress policies.
[References:, Kubernetes Docs — Network Policies: https://kubernetes.io/docs/concepts/services-networking/network-policies/,]

NEW QUESTION 17
Which of the following snippets from a RoleBinding correctly associates user bob with Role pod-reader ?

A. subjects:- kind: Username: pod-readerapiGroup: rbac.authorization.k8s.ioroleRef:kind: Rolename: bobapiGroup: rbac.authorization.k8s.io
B. subjects:- kind: Username: bobapiGroup: rbac.authorization.k8s.ioroleRef:kind: Rolename: pod-readerapiGroup: rbac.authorization.k8s.io
C. subjects:- kind: Username: bobapiGroup: rbac.authorization.k8s.ioroleRef:kind: ClusterRolename: pod-readerapiGroup: rbac.authorization.k8s.io
D. subjects:- kind: Groupname: bobapiGroup: rbac.authorization.k8s.ioroleRef:kind: Rolename: pod-readerapiGroup: rbac.authorization.k8s.io

Answer: B

Explanation:
Kubernetes RBAC usesRoleBindingto grant permissions defined in aRoleto asubject(user, group, or service account) within a namespace. The official example shows binding user jane to Role pod-reader:
"A RoleBinding grants the permissions defined in a Role to a user or set of users??."
Example:
subjects:
- kind: User
name: jane
apiGroup: rbac.authorization.k8s.io
roleRef:
kind: Role
name: pod-reader
apiGroup: rbac.authorization.k8s.io
— Kubernetes docs, RBAC: RoleBinding and ClusterRoleBinding
OptionBmatches this pattern exactly, with name: bob as theUsersubject and roleRef pointing to theRolenamed pod-reader.
Aswaps the names (subject is pod-reader, role is bob) ?? incorrect.
Creferences aClusterRole, not aRole(the question asks for Role).
Duses kind: Group even though we need theUserbob.
[References:, Kubernetes Docs — Using RBAC Authorization ??RoleBinding and ClusterRoleBinding: https://kubernetes.io/docs/reference/access-authn-authz/rbac/#rolebinding-and-clusterrolebinding,]

NEW QUESTION 22
How can a user enforce thePod Security Standardwithout third-party tools?

A. Through implementing Kyverno or OPA Policies.
B. Use the PodSecurity admission controller.
C. It is only possible to enforce the Pod Security Standard with additional tools within the cloud native ecosystem.
D. No additional measures have to be taken to enforce the Pod Security Standard.

Answer: B

Explanation:
ThePodSecurity admission controller(built-in as of Kubernetes v1.23+) enforces the Pod Security Standards (Privileged, Baseline, Restricted).
Enforcement is namespace-scoped and configured throughnamespace labels.
Incorrect options:
(A) Kyverno/OPA are external policy tools (useful but not required).
(C) Not true, PodSecurity admission provides native enforcement.
(D) Enforcement requires explicit configuration, not automatic.
[References:, Kubernetes Documentation – Pod Security Admission, CNCF Security Whitepaper – Policy enforcement and admission control.,]

NEW QUESTION 24
What is the purpose of the Supplier Assessments and Reviews control in the NIST 800-53 Rev. 5 set of controls for Supply Chain Risk Management?

A. To evaluate and monitor existing suppliers for adherence to security requirements.
B. To conduct regular audits of suppliers' financial performance.
C. To establish contractual agreements with suppliers.
D. To identify potential suppliers for the organization.

Answer: A

Explanation:
In NIST SP 800-53 Rev. 5,SR-6: Supplier Assessments and Reviewsrequires evaluating and monitoring suppliers' security and risk practices.
Exact extract (NIST SP 800-53 Rev. 5, SR-6):
"The organization assesses and monitors suppliers to ensure they are meeting the security requirements specified in contracts and agreements."
This is aboutongoing monitoringof supplier adherence, not financial audits, not contract creation, and not supplier discovery.
References:
NIST SP 800-53 Rev. 5, Control SR-6 (Supplier Assessments and Reviews): https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final

NEW QUESTION 27
What is the main reason an organization would use a Cloud Workload Protection Platform (CWPP) solution?

A. To protect containerized workloads from known vulnerabilities and malware threats.
B. To automate the deployment and management of containerized workloads.
C. To manage networking between containerized workloads in the Kubernetes cluster.
D. To optimize resource utilization and scalability of containerized workloads.

Answer: A

Explanation:
CWPP (Cloud Workload Protection Platform):As defined by Gartner and adopted across cloud security practices, CWPPs are designed tosecure workloads(VMs, containers, serverless functions) in hybrid and cloud environments.
They providevulnerability scanning, runtime protection, compliance checks, and malware detection.
Exact extract (Gartner CWPP definition):??Cloud workload protection platforms protect workloads regardless of location, including physical machines, VMs, containers, and serverless workloads. They provide vulnerability management, system integrity protection, intrusion detection and prevention, and malware protection.??

References:
Gartner: Cloud Workload Protection Platforms Market Guide (summary): https://www.gartner.com/reviews
/market/cloud-workload-protection-platforms
CNCF Security Whitepaper:https://github.com/cncf/tag-security

NEW QUESTION 29
By default, in a Kubeadm cluster, which authentication methods are enabled?

A. OIDC, Bootstrap tokens, and Service Account Tokens
B. X509 Client Certs, OIDC, and Service Account Tokens
C. X509 Client Certs, Bootstrap Tokens, and Service Account Tokens
D. X509 Client Certs, Webhook Authentication, and Service Account Tokens

Answer: C

Explanation:
In akubeadm cluster, by default the API server enables several authentication mechanisms:
X509 Client Certs: Used for authenticating kubelets, admins, and control-plane components.
Bootstrap Tokens: Temporary credentials used for node bootstrap/joining clusters.
Service Account Tokens: Used by workloads in pods to authenticate with the API server.
Exact extract (Kubernetes Docs – Authentication):
"Kubernetes uses client certificates, bearer tokens, an authenticating proxy, or HTTP basic auth to authenticate API requests."
"Bootstrap tokens are a simple bearer token that is meant to be used when creating new clusters or joining new nodes to an existing cluster."
"Service accounts are special accounts that provide an identity for processes that run in a Pod."
References:
Kubernetes Docs — Authentication: https://kubernetes.io/docs/reference/access-authn-authz/authentication/
Kubeadm — TLS Bootstrapping: https://kubernetes.io/docs/reference/access-authn-authz/bootstrap-tokens/

NEW QUESTION 32
.....

THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual KCSA Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the KCSA Product From:

https://www.2passeasy.com/dumps/KCSA/

Money Back Guarantee

KCSA Practice Exam Features:

* KCSA Questions and Answers Updated Frequently

* KCSA Practice Questions Verified by Expert Senior Certified Staff

* KCSA Most Realistic Questions that Guarantee you a Pass on Your FirstTry

* KCSA Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year