Linux-Foundation

Exam Questions KCSA

Kubernetes and Cloud Native Security Associate (KCSA)

NEW QUESTION 1
Which standard approach to security is augmented by the 4C??s of Cloud Native security?

A. Zero Trust
B. Least Privilege
C. Defense-in-Depth
D. Secure-by-Design

Answer: C

Explanation:

The 4C??s model (Cloud, Cluster, Container, Code) is presented in the official Kubernetes documentation as alayeredmodel that explicitly maps todefense-in-depth.

Exact extracts from Kubernetes docs(security overview):

??The 4C??s of Cloud Native Security are Cloud, Clusters, Containers, and Code.??

??You can think of the 4C??s asa layered approach to security; applying security measures at each layer reduces risk.??

??This layered approach is commonly known asdefense in depth.??

References:
Kubernetes Docs — Security overview #The 4C??s of Cloud Native Security: https://kubernetes.io/docs /concepts/security/overview/#the-4cs-of-cloud-native-security

NEW QUESTION 2
Given a standard Kubernetes cluster architecture comprising a single control plane node (hosting bothetcdand the control plane as Pods) and three worker nodes, which of the following data flows crosses atrust boundary?

A. From kubelet to Container Runtime
B. From kubelet to API Server
C. From kubelet to Controller Manager
D. From API Server to Container Runtime

Answer: B

Explanation:

Trust boundariesexist where data flows between different security domains.

In Kubernetes:

Communication between thekubelet (node agent)and theAPI Server (control plane)crosses the node-to-control-plane trust boundary.

(A) Kubelet to container runtime is local, no boundary crossing.

(C) Kubelet does not communicate directly with the controller manager.

(D) API server does not talk directly to the container runtime; it delegates to kubelet.

Therefore, (B) is the correct trust boundary crossing flow.

References:
CNCF Security Whitepaper – Kubernetes Threat Model: identifies node-to-control-plane communications (kubelet # API Server) as crossing trust boundaries.
Kubernetes Documentation – Cluster Architecture

NEW QUESTION 3
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 4
An attacker has successfully overwhelmed the Kubernetes API server in a cluster with a single control plane node by flooding it with requests.
How would implementing a high-availability mode with multiple control plane nodes mitigate this attack?

A. By implementing network segmentation to isolate the API server from the rest of the cluster, preventing the attack from spreading.
B. By distributing the workload across multiple API servers, reducing the load on each server.
C. By increasing the resources allocated to the API server, allowing it to handle a higher volume of requests.
D. By implementing rate limiting and throttling mechanisms on the API server to restrict the number of requests allowed.

Answer: B

Explanation:

Inhigh-availability clusters, multiple API server instances run behind a load balancer.

This distributes client requests across multiple API servers, preventing a single API server from being overwhelmed.

Exact extract (Kubernetes Docs – High Availability Clusters):
"A highly available control plane runs multiple instances of kube-apiserver, typically fronted by a load balancer, so that if one instance fails or is overloaded, others continue serving requests."

Other options clarified:
A: Network segmentation does not directly mitigate API server DoS.
C: Adding resources helps, but doesn??t solve single-point-of-failure.
D: Rate limiting is a valid mitigation but not provided by HA alone.
[References:, Kubernetes Docs — Building High-Availability Clusters: https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/high-availability/,]

NEW QUESTION 5
Is it possible to restrict permissions so that a controller can only change the image of a deployment (without changing anything else about it, e.g., environment variables, commands, replicas, secrets)?

A. Yes, by granting permission to the /image subresource.
B. Not with RBAC, but it is possible with an admission webhook.
C. No, because granting access to the spec.containers.image field always grants access to the rest of the spec object.
D. Yes, with a 'managed fields' annotation.

Answer: B

Explanation:
RBAC in Kubernetesis coarse-grained: it controlsverbs(get, update, patch, delete) onresources(e.g., deployments), butnot individual fieldswithin a resource.
There isno /image subresource for deployments(there is one for pods but only for ephemeral containers).
Therefore,RBAC cannot restrict changes only to the image field.
Admission Webhooks(mutating/validating)canenforce fine-grained policies (e.g., deny updates that change anything other than spec.containers[*].image).
Exact extract (Kubernetes Docs – Admission Webhooks):
"Admission webhooks can be used to enforce custom policies on objects being admitted."
[References:, Kubernetes Docs — RBAC: https://kubernetes.io/docs/reference/access-authn-authz/rbac/, Kubernetes Docs — Admission Webhooks: https://kubernetes.io/docs/reference/access-authn-authz/extensible-admission-controllers/,]

NEW QUESTION 6
You want to minimize security issues in running Kubernetes Pods. Which of the following actions can help achieve this goal?

A. Sharing sensitive data among Pods in the same cluster to improve collaboration.
B. Running Pods with elevated privileges to maximize their capabilities.
C. Implement Pod Security standards in the Pod's YAML configuration.
D. Deploying Pods with randomly generated names to obfuscate their identities.

Answer: C

Explanation:

Pod Security Standards (PSS):
Kubernetes providesPod Security Admission (PSA)to enforce security controls based on policies.
Official extract: ??Pod Security Standards define different isolation levels for Pods. The standards focus on restricting what Pods can do and what they can access.??
The three standard profiles are:

Privileged: unrestricted (not recommended).
Baseline: minimal restrictions.
Restricted: highly restricted, enforcing least privilege.

Why option C is correct:
Applying Pod Security Standards in YAML ensures Pods adhere tobest practiceslike:
No root user.
Restricted host access.
No privilege escalation.
Seccomp/AppArmor profiles.
This directly minimizes security risks.

Why others are wrong:
A:Sharing sensitive data increases risk of exposure.
B:Running with elevated privileges contradicts least privilege principle.
D:Random Pod names donotcontribute to security.
[References:, Kubernetes Docs — Pod Security Standards: https://kubernetes.io/docs/concepts/security/pod-security-standards/, Kubernetes Docs — Pod Security Admission: https://kubernetes.io/docs/concepts/security/pod-security-admission/,]

NEW QUESTION 7
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 8
A container running in a Kubernetes cluster has permission to modify host processes on the underlying node.
What combination of privileges and capabilities is most likely to have led to this privilege escalation?

A. There is no combination of privileges and capabilities that permits this.
B. hostPID and SYS_PTRACE
C. hostPath and AUDIT_WRITE
D. hostNetwork and NET_RAW

Answer: B

Explanation:
hostPID:When enabled, the container shares the host??s process namespace ?? container can see and potentially interact with host processes.
SYS_PTRACE capability:Grants the container the ability to trace, inspect, and modify other processes (e.g., via ptrace).
Combination of hostPID + SYS_PTRACE allows a container toattach to and modify host processes, which is a direct privilege escalation.
Other options explained:
hostPath + AUDIT_WRITE:hostPath exposes filesystem paths but does not inherently allow process modification.
hostNetwork + NET_RAW:grants raw socket access but only for networking, not host process modification.
A:Incorrect — such combinationsdo exist(like B).
[References:, Kubernetes Docs — Configure a Pod to use hostPID: https://kubernetes.io/docs/tasks/configure-pod-container/share-process-namespace/, Linux Capabilities man page: https://man7.org/linux/man-pages/man7/capabilities.7.html,]

NEW QUESTION 9
What was the name of the precursor to Pod Security Standards?

A. Container Runtime Security
B. Kubernetes Security Context
C. Container Security Standards
D. Pod Security Policy

Answer: D

Explanation:
Kubernetes originally had a feature calledPodSecurityPolicy (PSP), which provided controls to restrict pod behavior.
Official docs:
"PodSecurityPolicy was deprecated in Kubernetes v1.21 and removed in v1.25."
"Pod Security Standards (PSS) replace PodSecurityPolicy (PSP) with a simpler, policy-driven approach."
PSP was often complex and hard to manage, so it was replaced by Pod Security Admission (PSA) which enforcesPod Security Standards.
[References:, Kubernetes Docs — PodSecurityPolicy (deprecated): https://kubernetes.io/docs/concepts/security/pod-security-policy/, Kubernetes Blog — PodSecurityPolicy Deprecation: https://kubernetes.io/blog/2021/04/06/podsecuritypolicy-deprecation-past-present-and-future/,]

NEW QUESTION 10
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 13
Which of the following statements on static Pods is true?

A. The kubelet can run static Pods that span multiple nodes, provided that it has the necessary privileges from the API server.
B. The kubelet can run a maximum of 5 static Pods on each node.
C. The kubelet schedules static Pods local to its node without going through the kube-scheduler, making tracking and managing them difficult.
D. The kubelet only deploys static Pods when the kube-scheduler is unresponsive.

Answer: C

Explanation:
Static Podsare managed directly by thekubeleton each node.
They arenot scheduled by the kube-schedulerand always remain bound to the node where they are defined.
Exact extract (Kubernetes Docs – Static Pods):
??Static Pods are managed directly by the kubelet daemon on a specific node, without the API server. They do not go through the Kubernetes scheduler.??
Clarifications:
A: Static Pods do not span multiple nodes.
B: No hard limit of 5 Pods per node.
D: They are not a fallback mechanism; kubelet always manages them regardless of scheduler state.
References:
Kubernetes Docs — Static Pods: https://kubernetes.io/docs/tasks/configure-pod-container/static-pod/

NEW QUESTION 18
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 20
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 21
A container image istrojanizedby an attacker by compromising the build server. Based on the STRIDE threat modeling framework, which threat category best defines this threat?

A. Repudiation
B. Spoofing
C. Denial of Service
D. Tampering

Answer: D

Explanation:
In STRIDE,Tamperingis the threat category forunauthorized modification of data or code/artifacts. A trojanized container image is, by definition, an attacker??smodificationof the build output (the image) after compromising the CI/build system—i.e., tampering with the artifact in the software supply chain.
Why not the others?
Spoofingis about identity/authentication (e.g., pretending to be someone/something).
Repudiationis about denying having performed an action without sufficient audit evidence.
Denial of Servicetargets availability (exhausting resources or making a service unavailable).The scenario explicitly focuses on analtered imageresulting from a compromised build server—this squarely maps toTampering.
Authoritative references (for verification and deeper reading):
Kubernetes (official docs)– Supply Chain Security (discusses risks such as compromised CI/CD pipelines leading to modified/poisoned images and emphasizes verifying image integrity/signatures).
Kubernetes Docs#Security#Supply chain securityandSecuring a cluster(sections on image provenance, signing, and verifying artifacts).
CNCF TAG Security – Cloud Native Security Whitepaper (v2)– Threat modeling in cloud-native and software supply chain risks; describes attackers modifying build outputs (images/artifacts) via CI
/CD compromise as a form oftamperingand prescribes controls (signing, provenance, policy).
CNCF TAG Security – Software Supply Chain Security Best Practices– Explicitly covers CI/CD compromise leading tomaliciously modified imagesand recommends SLSA, provenance attestation, and signature verification (policy enforcement via admission controls).
Microsoft STRIDE (canonical reference)– DefinesTamperingasmodifying data or code, which directly fits a trojanized image produced by a compromised build system.

NEW QUESTION 25
How do Kubernetes namespaces impact the application of policies when using Pod Security Admission?

A. Namespaces are ignored; Pod Security Admission policies apply cluster-wide only.
B. Different policies can be applied to specific namespaces.
C. Each namespace can have only one active policy.
D. The default namespace enforces the strictest security policies by default.

Answer: B

Explanation:
Pod Security Admission (PSA)enforces policies by applyinglabels on namespaces, not globally across the cluster.

Exact extract (Kubernetes Docs – Pod Security Admission):
??You can apply Pod Security Standards to namespaces by adding labels such as pod- security.kubernetes.io/enforce. Different namespaces can enforce different policies.??
Clarifications:
A: Incorrect, namespaces are the unit of enforcement.
C: Misleading — a namespace can have multiple enforcement modes (enforce, audit, warn).
D: Default namespace doesnotenforce strict policies unless labeled.
References:
Kubernetes Docs — Pod Security Admission: https://kubernetes.io/docs/concepts/security/pod-security- admission/

NEW QUESTION 29
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 30
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 31
What is Grafana?

A. A cloud-native distributed tracing system for monitoring microservices architectures.
B. A container orchestration platform for managing and scaling applications.
C. A platform for monitoring and visualizing time-series data.
D. A cloud-native security tool for scanning and detecting vulnerabilities in Kubernetes clusters.

Answer: C

Explanation:
Grafana:An open-source analytics and visualization platform widely used with Prometheus, Loki, etc.
Exact extract (Grafana Docs):??Grafana is the open-source analytics and monitoring solution for every database. It allows you to query, visualize, alert on, and understand your metrics no matter where they are stored.??
A is wrong:That describesJaeger(distributed tracing).
B is wrong:That??sKubernetesitself.
D is wrong:That??sTrivy/Aqua/Prismatype tools.
References:
Grafana Docs: https://grafana.com/docs/grafana/latest/

NEW QUESTION 36
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 41
Which of the following is a control for Supply Chain Risk Management according to NIST 800-53 Rev. 5?

A. Access Control
B. System and Communications Protection
C. Supply Chain Risk Management Plan
D. Incident Response

Answer: C

NEW QUESTION 43
.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questons and Answers in PDF Format

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