Linux-Foundation

Exam Questions KCSA

Kubernetes and Cloud Native Security Associate (KCSA)

NEW QUESTION 1
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 2
Which of the following represents a baseline security measure for containers?

A. Implementing access control to restrict container access.
B. Configuring a static IP for each container.
C. Configuring persistent storage for containers.
D. Run containers as the root user.

Answer: A

Explanation:

Access control (RBAC, least privilege, user restrictions)is abaseline container security best practice.

Exact extract (Kubernetes Pod Security Standards – Baseline):

??The baseline profile is designed to prevent known privilege escalations. It prohibits running privileged containers or containers as root.??

Other options clarified:

B: Static IPs not a security measure.

C: Persistent storage is functionality, not security.

D: Running as root is explicitlyinsecure.
References:
Kubernetes Docs — Pod Security Standards (Baseline): https://kubernetes.io/docs/concepts/security/pod- security-standards/

NEW QUESTION 3
Which of the following statements best describes the role of the Scheduler in Kubernetes?

A. The Scheduler is responsible for monitoring and managing the health of the Kubernetes cluster.
B. The Scheduler is responsible for ensuring the security of the Kubernetes cluster and its components.
C. The Scheduler is responsible for managing the deployment and scaling of applications in the Kubernetes cluster.
D. The Scheduler is responsible for assigning Pods to nodes based on resource availability and other constraints.

Answer: D

Explanation:

TheKubernetes Schedulerassigns Pods to nodes based on:

Resource requests & availability (CPU, memory, GPU, etc.)

Constraints (affinity, taints, tolerations, topology, policies)

Exact extract (Kubernetes Docs – Scheduler):

??The scheduler is a control plane process that assigns Pods to Nodes. Scheduling decisions take into account resource requirements, affinity/anti-affinity, constraints, and policies.??

Other options clarified:

A: Monitoring cluster health is theController Manager??s/kubelet??s job.

B: Security is enforced throughRBAC, admission controllers, PSP/PSA, not the scheduler.

C: Deployment scaling is handled by theController Manager(Deployment/ReplicaSet controller).

References:
Kubernetes Docs — Scheduler: https://kubernetes.io/docs/concepts/scheduling-eviction/kube-scheduler/

NEW QUESTION 4
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 5
Why does the defaultbase64 encodingthat Kubernetes applies to the contents of Secret resources provide inadequate protection?

A. Base64 encoding is vulnerable to brute-force attacks.
B. Base64 encoding relies on a shared key which can be easily compromised.
C. Base64 encoding does not encrypt the contents of the Secret, only obfuscates it.
D. Base64 encoding is not supported by all Secret Stores.

Answer: C

Explanation:

Kubernetes stores Secret data asbase64-encoded stringsin etcd by default.

Base64 is not encryption— it is a simple encoding scheme that merelyobfuscatesdata for transport and storage. Anyone with read access to etcd or the Secret manifest can easily decode the value back to plaintext.

For actual protection, Kubernetes supportsencryption at rest(via encryption providers) and external Secret management (Vault, KMS, etc.).
[References:, Kubernetes Documentation – Secrets, CNCF Security Whitepaper – Data protection section: highlights that base64 encoding does not protect data and encryption at rest is recommended.,]

NEW QUESTION 6
Which other controllers are part of the kube-controller-manager inside the Kubernetes cluster?

A. Job controller, CronJob controller, and DaemonSet controller
B. Pod, Service, and Ingress controller
C. Namespace controller, ConfigMap controller, and Secret controller
D. Replication controller, Endpoints controller, Namespace controller, and ServiceAccounts controller

Answer: D

Explanation:

kube-controller-managerruns a set of controllers that regulate the cluster??s state.

Exact extract (Kubernetes Docs): "The kube-controller-manager runs controllers that are core to Kubernetes. Examples of controllers are: Node controller, Replication controller, Endpoints controller, Namespace controller, and ServiceAccounts controller."

Why D is correct:All listed are actual controllers within kube-controller-manager.

Why others are wrong:

A:Job and CronJob controllers are managed by kube-controller-manager, but DaemonSet controller is managed by the kube-scheduler/deployment logic.

B:Pod, Service, Ingress controllers are not part of kube-controller-manager.

C:ConfigMap and Secret do not have dedicated controllers.

[References:, Kubernetes Docs — kube-controller-manager: https://kubernetes.io/docs/reference/command-line-tools-reference/kube-controller-manager/,]

NEW QUESTION 7
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 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
Which of the following statements correctly describes a container breakout?

A. A container breakout is the process of escaping the container and gaining access to the Pod's network traffic.
B. A container breakout is the process of escaping a container when it reaches its resource limits.
C. A container breakout is the process of escaping the container and gaining access to the cloud provider's infrastructure.
D. A container breakout is the process of escaping the container and gaining access to the host operating system.

Answer: D

Explanation:
Container breakoutrefers to an attacker escaping container isolation and reaching thehost OS.
Once the host is compromised, the attacker can accessother containers, Kubernetes nodes, or escalate further.
Exact extract (Kubernetes Security Docs):
??If an attacker gains access to a container, they may attempt a container breakout to gain access to the host system.??

Other options clarified:
A: Network access inside a Pod # breakout.
B: Resource exhaustion is aDoS, not a breakout.
C: Cloud infrastructure compromise is possibleafterhost compromise, but not the definition of breakout.
References:
Kubernetes Security Concepts: https://kubernetes.io/docs/concepts/security/
CNCF Security Whitepaper (Threats section):https://github.com/cncf/tag-security

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 10
When should soft multitenancy be used over hard multitenancy?

A. When the priority is enabling resource sharing and efficiency between tenants.
B. When the priority is enabling complete isolation between tenants.
C. When the priority is enabling fine-grained control over tenant resources.
D. When the priority is enabling strict security boundaries between tenants.

Answer: A

Explanation:
Soft multitenancy(Namespaces, RBAC, Network Policies) # assumes some level of trust between tenants, focuses onresource sharing and efficiency.
Hard multitenancy(separate clusters or strong virtualization) # strict isolation, used when tenants are untrusted.
Exact extract (CNCF TAG Security Multi-Tenancy Whitepaper):
??Soft multi-tenancy refers to multiple workloads running in the same cluster with some trust assumptions. It provides resource sharing and operational efficiency.
Hard multi- tenancy requires stronger isolation guarantees, typically separate clusters.??
References:
CNCF Security TAG — Multi-Tenancy Whitepaper:https://github.com/cncf/tag-security/tree/main/multi- tenancy

NEW QUESTION 14
In which order are thevalidating and mutating admission controllersrun while the Kubernetes API server processes a request?

A. The order of execution varies and is determined by the cluster configuration.
B. Validating admission controllers run before mutating admission controllers.
C. Validating and mutating admission controllers run simultaneously.
D. Mutating admission controllers run before validating admission controllers.

Answer: D

Explanation:
Theadmission control flowin Kubernetes:
Mutating admission controllersrun first and can modify incoming requests.
Validating admission controllersrun after mutations to ensure the final object complies with policies.
This ensures policies validate thefinal, mutated object.
References:
Kubernetes Documentation – Admission Controllers CNCF Security Whitepaper – Admission control workflow.

NEW QUESTION 17
In a cluster that contains Nodes withmultiple container runtimesinstalled, how can a Pod be configured to be created on a specific runtime?

A. By using a command-line flag when creating the Pod.
B. By modifying the Docker daemon configuration.
C. By setting the container runtime as an environment variable in the Pod.
D. By specifying the container runtime in the Pod's YAML file.

Answer: D

Explanation:
Kubernetes supportsmultiple container runtimeson a node via theRuntimeClassresource.
To select a runtime, you specify the runtimeClassName field in thePod??s YAML manifest. Example:
apiVersion: v1
kind: Pod
metadata:
name: example
spec:
runtimeClassName: gvisor
containers:
- name: app

image: nginx
Incorrect options:
(A) You cannot specify container runtime through a kubectl command-line flag.
(B) Modifying the Docker daemon config does not direct Kubernetes Pods to a runtime.
(C) Environment variables inside a Pod spec do not control container runtimes.
References:
Kubernetes Documentation – RuntimeClass
CNCF Security Whitepaper – Workload isolation via different runtimes (e.g., gVisor, Kata) for enhanced security.

NEW QUESTION 21
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 24
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 29
To restrict the kubelet's rights to the Kubernetes API, whatauthorization modeshould be set on the Kubernetes API server?

A. Node
B. AlwaysAllow
C. kubelet
D. Webhook

Answer: A

Explanation:
TheNode authorization modeis designed to specifically limit what kubelets can do when they connect to the Kubernetes API server.
It authorizes requests from kubelets based on the Pods scheduled to run on their nodes, ensuring kubelets cannot interact with resources beyond their scope.
Incorrect options:
(B)AlwaysAllowallows unrestricted access (insecure).
(C) No kubelet authorization mode exists.
(D)Webhookmode delegates authorization decisions to an external service, not specifically for kubelets.
[References:, Kubernetes Documentation – Node Authorization, CNCF Security Whitepaper – Access control: kubelet authorization and Node authorizer.,]

NEW QUESTION 30
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 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 32
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 34
Which step would give an attacker a foothold in a cluster butno long-term persistence?

A. Modify Kubernetes objects stored within etcd.
B. Modify file on host filesystem.
C. Starting a process in a running container.
D. Create restarting container on host using Docker.

Answer: C

Explanation:
Starting a process in a running containerprovides an attacker withtemporary execution (foothold)inside the cluster, but once the container is stopped or restarted, that malicious process is lost. This means the attacker has nolong-term persistence.
Incorrect options:
(A) Modifying objects inetcdgrants persistent access since cluster state is stored in etcd.
(B) Modifying files on thehost filesystemcan create persistence across reboots or container restarts.
(D) Creating a restarting container directly on the host via Docker bypasses Kubernetes but persists across pod restarts if Docker restarts it.
[References:, CNCF Security Whitepaper – Threat Modeling section: Describes howephemeral processes inside containersprovide attackers short-term control but not durable persistence., Kubernetes Documentation – Cluster Threat Model emphasizes ephemeral vs. persistent attacker footholds.,]

NEW QUESTION 36
.....

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