LinuxFoundation.KCSA.v2026-03-21.q21

Exam Code:	KCSA
Exam Name:	Linux Foundation Kubernetes and Cloud Native Security Associate
Certification Provider:	Linux Foundation
Free Question Number:	21
Version:	v2026-03-21
# of views:	118
# of Questions views:	210
https://www.freeqas.com/qa/Linux-Foundation/KCSA/LinuxFoundation.KCSA.v2026-03-21.q21.html	

NEW QUESTION: 1
A user runs a command with kubectl to apply a change to a deployment. What is the first Kubernetes component that the request reaches?
A. Kubernetes Controller Manager
B. Kubernetes API Server
C. Kubernetes Scheduler
D. kubelet
Answer: B (LEAVE A REPLY)
* Allkubectl requestsgo to theKubernetes API Server.
* The API server is thefront-end of the control planeand validates/authenticates requests before other components act.
* Exact extract (Kubernetes Docs - Components):
* "The API server is a component of the Kubernetes control plane that exposes the Kubernetes API. It is the front end for the Kubernetes control plane."
* Other options clarified:
* Controller Manager: reconciles state after API Server processes the request.
* Scheduler: assigns Pods to nodes after API Server accepts workload objects.
* kubelet: node agent, only communicates after API Server updates desired state.
References:
Kubernetes Docs - Components:
https://kubernetes.io/docs/concepts/overview/components/

NEW QUESTION: 2
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 (LEAVE A REPLY)
* 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: 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 (LEAVE A REPLY)
* 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
An attacker has access to the network segment that the cluster is on.
What happens when a compromised Pod attempts to connect to the API server?
A. The compromised Pod is automatically isolated from the network to prevent any connections to the API server.
B. The compromised Pod is allowed to connect to the API server without any restrictions.
C. The compromised Pod attempts to connect to the API server, but its requests may be blocked due to network policies.
D. The compromised Pod connects to the API server and is granted elevated privileges by default.
Answer: C (LEAVE A REPLY)
* By default,Pods can connect to the API server(since ServiceAccount tokens are mounted).
* However, whether they succeed in acting depends on:
* Network Policies(may block egress).
* RBAC(controls permissions).
* Exact extract (Kubernetes Docs - API Access):
* "Pods authenticate to the API server using the service account token mounted into the Pod.
Authorization is then enforced by RBAC. NetworkPolicies may further restrict access."
* Clarifications:
* A: No default automatic isolation.
* B: Not always unrestricted; policies may apply.
* D: Pods get minimal default privileges, not automatic elevation.
References:
Kubernetes Docs - API Access to Pods:
https://kubernetes.io/docs/concepts/security/service-accounts/ Kubernetes Docs - Network Policies: https://kubernetes.io/docs/concepts/services-networking/network- policies/

NEW QUESTION: 5
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 (LEAVE A REPLY)
* 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: 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: (SHOW ANSWER)
* 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
Which way of defining security policy brings consistency, minimizes toil, and reduces the probability of misconfiguration?
A. Using a declarative approach to define security policies as code.
B. Relying on manual audits and inspections for security policy enforcement.

C. Manually configuring security controls for each individual resource, regularly.
D. Implementing security policies through manual scripting on an ad-hoc basis.
Answer: A (LEAVE A REPLY)
* Defining policiesas code (declarative)is a best practice in Kubernetes and cloud-native security.
* This is aligned withGitOpsandPolicy-as-Codeprinciples (OPA Gatekeeper, Kyverno, etc.).
* Exact extract (CNCF Security Whitepaper):
* "Policy-as-Code enables declarative definition and enforcement of security policies, bringing consistency, automation, and reducing misconfiguration risk."
* Manual audits, ad-hoc scripting, or individual configurations are error-prone and inconsistent.
References:
CNCF Security Whitepaper:https://github.com/cncf/tag-security
Kubernetes Docs - Policy as Code (OPA, Kyverno):
https://kubernetes.io/docs/concepts/security/

NEW QUESTION: 8
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 (LEAVE A REPLY)
* 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: 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 (LEAVE A REPLY)
* 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
In order to reduce the attack surface of the Scheduler, which default parameter should be set to false?
A. --scheduler-name
B. --profiling
C. --secure-kubeconfig
D. --bind-address
Answer: B (LEAVE A REPLY)
* Thekube-schedulerexposes aprofiling/debugging endpointwhen --profiling=true (default).
* This can unnecessarily increase the attack surface.
* Best practice: set --profiling=false in production.
* Exact extract (Kubernetes Docs - kube-scheduler flags):
* "--profiling (default true): Enable profiling via web interface host:port/debug/pprof/."
* Why others are wrong:

* --scheduler-name: just identifies the scheduler, not a security risk.
* --secure-kubeconfig: not a valid flag.
* --bind-address: changing it limits exposure but is not the default risk parameter for profiling.
References:
Kubernetes Docs - kube-scheduler options:
https://kubernetes.io/docs/reference/command-line-tools- reference/kube-scheduler/

NEW QUESTION: 11
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 (LEAVE A REPLY)
* 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: 12
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: (SHOW ANSWER)
* NIST SP 800-53 Rev. 5 introduces a dedicated family of controls calledSupply Chain Risk Management (SR).
* Within SR,SR-2 (Supply Chain Risk Management Plan)is a specific control.
* Exact extract from NIST 800-53 Rev. 5:
* "The organization develops and implements a supply chain risk management plan for the system, system component, or system service."
* While Access Control, System and Communications Protection, and Incident Response are control families, the correctsupply chain-specific controlis theSupply Chain Risk Management Plan (SR-2).
References:
NIST SP 800-53 Rev. 5 -Security and Privacy Controls for Information Systems and Organizations:
https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final

NEW QUESTION: 13
In Kubernetes, what isPublic Key Infrastructure (PKI)used for?
A. To manage certificates and ensure secure communication in a Kubernetes cluster.
B. To automate the scaling of containers in a Kubernetes cluster.
C. To manage networking in a Kubernetes cluster.
D. To monitor and analyze performance metrics of a Kubernetes cluster.
Answer: (SHOW ANSWER)
* Kubernetes usesPKI certificatesextensively to secure communication between control plane components (API server, etcd, kube-scheduler, kube-controller-manager) and with kubelets.
* Certificates enablemutual TLS authentication and encryptionacross components.
* PKI does not handle scaling, networking, or monitoring.
References:
Kubernetes Documentation - Certificates
CNCF Security Whitepaper - Cluster communication security and the role of PKI.

NEW QUESTION: 14
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: (SHOW ANSWER)

* 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: 15
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: (SHOW ANSWER)
* 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: 16
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 (LEAVE A REPLY)
* 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: 17
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 managing the deployment and scaling of applications in the Kubernetes cluster.
C. The Scheduler is responsible for ensuring the security of the Kubernetes cluster and its components.
D. The Scheduler is responsible for assigning Pods to nodes based on resource availability and other constraints.
Answer: (SHOW ANSWER)
* 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: 18
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: (SHOW ANSWER)
* 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: 19
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 (LEAVE A REPLY)
* 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: 20
Why mightNetworkPolicyresources have no effect in a Kubernetes cluster?
A. NetworkPolicy resources are only enforced if the Kubernetes scheduler supports them.
B. NetworkPolicy resources are only enforced if the networking plugin supports them.
C. NetworkPolicy resources are only enforced for unprivileged Pods.
D. NetworkPolicy resources are only enforced if the user has the right RBAC permissions.

Answer: (SHOW ANSWER)
* NetworkPolicies define how Pods can communicate with each other and external endpoints.
* However, Kubernetes itselfdoes not enforce NetworkPolicy. Enforcement depends on theCNI plugin used (e.g., Calico, Cilium, Kube-Router, Weave Net).
* If a cluster is using a network plugin that does not support NetworkPolicies, then creating NetworkPolicy objects hasno effect.
References:
Kubernetes Documentation - Network Policies
CNCF Security Whitepaper - Platform security section: notes that security enforcement relies on CNI capabilities.

NEW QUESTION: 21
An attacker compromises a Pod and attempts to use its service account token to escalate privileges within the cluster. Which Kubernetes security feature is designed tolimit what this service account can do?
A. PodSecurity admission
B. NetworkPolicy
C. Role-Based Access Control (RBAC)
D. RuntimeClass
Answer: C (LEAVE A REPLY)
* When a Pod is created, Kubernetes automatically mounts aservice account tokenthat can authenticate to the API server.
* TheRole-Based Access Control (RBAC)system defines what actions a service account can perform.
* By carefully restricting Roles and RoleBindings, administrators limit the blast radius of a compromised Pod.
* Incorrect options:
* (A)PodSecurity admissionenforces workload-level security settings but does not control API access.
* (B)NetworkPolicycontrols network communication, not API privileges.
* (D)RuntimeClassselects container runtimes, unrelated to privilege escalation through API tokens.
References:
Kubernetes Documentation - Using RBAC Authorization
CNCF Security Whitepaper - Identity & Access Management: limiting lateral movement by constraining service account permissions.