Linux-Foundation

Exam Questions KCSA

Kubernetes and Cloud Native Security Associate (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 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 3
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 4
On a client machine, what directory (by default) contains sensitive credential information?

A. /etc/kubernetes/
B. $HOME/.kube
C. /opt/kubernetes/secrets/
D. $HOME/.config/kubernetes/

Answer: B

Explanation:

- Thekubectlclient uses configuration from$HOME/.kube/configby default.
- This file contains: cluster API server endpoint, user certificates, tokens, or kubeconfigs #sensitive credentials.
- Exact extract (Kubernetes Docs – Configure Access to Clusters):
- ??By default, kubectl looks for a file named config in the $HOME/.kube directory. This file contains configuration information including user credentials.??
- Other options clarified:
- A: /etc/kubernetes/ exists on nodes (control plane) not client machines.
- C: /opt/kubernetes/secrets/ is not a standard path.
- D: $HOME/.config/kubernetes/ is not where kubeconfig is stored by default.

References:
Kubernetes Docs — Configure Access to Clusters: https://kubernetes.io/docs/concepts/configuration/organize- cluster-access-kubeconfig/

NEW QUESTION 5
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 6
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: A

Explanation:
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 7
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 8
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 9
A cluster administrator wants to enforce the use of a different container runtime depending on the application a workload belongs to.

A. By manually modifying the container runtime for each workload after it has been created.
B. By modifying the kube-apiserver configuration file to specify the desired container runtime for each application.
C. By configuring avalidating admission controllerwebhook that verifies the container runtime based on the application label and rejects requests that do not comply.
D. By configuring amutating admission controllerwebhook that intercepts new workload creation requests and modifies the container runtime based on the application label.

Answer: D

Explanation:

Kubernetes supports workload-specific runtimes viaRuntimeClass.

Amutating admission controllercan enforce this automatically by:

Intercepting workload creation requests.

Modifying the Pod spec to set runtimeClassName based on labels or policies.

Incorrect options:
(A) Manual modification is not scalable or secure.
(B) kube-apiserver cannot enforce per-application runtime policies.
(C) A validating webhook can onlyreject, not modify, the runtime.
[References:, Kubernetes Documentation – RuntimeClass, CNCF Security Whitepaper – Admission controllers for enforcing runtime policies.,]

NEW QUESTION 10
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 10
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 12
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 15
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 19
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

Explanation:
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 24
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 26
As a Kubernetes and Cloud Native Security Associate, a user can set upaudit loggingin a cluster. What is the risk of logging every event at the fullRequestResponselevel?

A. No risk, as it provides the most comprehensive audit trail.
B. Increased storage requirements and potential impact on performance.
C. Improved security and easier incident investigation.
D. Reduced storage requirements and faster performance.

Answer: B

Explanation:
Audit loggingrecords API server requests and responses for security monitoring.
TheRequestResponse levellogs the full request and response bodies, which can:
Significantly increasestorage and performance overhead.
Potentially log sensitive data (including Secrets).
Therefore, while comprehensive, it introduces risks of performance degradation and excessive log volume.
References:
Kubernetes Documentation – Auditing
CNCF Security Whitepaper – Logging and monitoring: trade-offs between verbosity, storage, and security.

NEW QUESTION 27
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 30
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: B

Explanation:
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 33
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 38
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 39
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

Explanation:
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.

NEW QUESTION 42
What kind of organization would need to be compliant with PCI DSS?

A. Retail stores that only accept cash payments.
B. Government agencies that collect personally identifiable information.
C. Non-profit organizations that handle sensitive customer data.
D. Merchants that process credit card payments.

Answer: D

Explanation:
PCI DSS (Payment Card Industry Data Security Standard):applies to any entity thatstores, processes, or transmits cardholder data.
Exact extract (PCI DSS official summary):
"PCI DSS applies to all entities that store, process or transmit cardholder data (CHD) and /or sensitive authentication data (SAD)."
Therefore,merchants who process credit card paymentsmust comply.
Why others are wrong:
A: No card payments, so no PCI scope.
B: This falls underFISMA / NIST 800-53, not PCI DSS.
C: Non-profits may handle sensitive data, but PCI only applies if they processcredit cards.
References:
PCI Security Standards Council — PCI DSS Summary: https://www.pcisecuritystandards.org/pci_security/

NEW QUESTION 47
Which security knowledge-base focuses specifically onoffensive tools, techniques, and procedures?

A. MITRE ATT&CK
B. OWASP Top 10
C. CIS Controls
D. NIST Cybersecurity Framework

Answer: A

Explanation:
MITRE ATT&CKis a globally recognizedknowledge base of adversary tactics, techniques, and procedures (TTPs). It is focused on describingoffensive

behaviorsattackers use.
Incorrect options:
(B)OWASP Top 10highlights common application vulnerabilities, not attacker techniques.
(C)CIS Controlsare defensive best practices, not offensive tools.
(D)NIST Cybersecurity Frameworkprovides a risk-based defensive framework, not adversary TTPs.
References:
MITRE ATT&CK Framework
CNCF Security Whitepaper – Threat intelligence section: references MITRE ATT&CK for describing attacker behavior.

NEW QUESTION 50
You are responsible for securing thekubeletcomponent in a Kubernetes cluster.
Which of the following statements about kubelet security is correct?

A. Kubelet runs as a privileged container by default.
B. Kubelet does not have any built-in security features.
C. Kubelet supports TLS authentication and encryption for secure communication with the API server.
D. Kubelet requires root access to interact with the host system.

Answer: C

Explanation:
Thekubeletis the primary agent that runs on each node in a Kubernetes cluster and communicates with the control plane.
Kubeletsupports TLS (Transport Layer Security)for both authentication and encryption when interacting with the API server. This is a core security feature that ensures secure node-to-control-plane communication.
Incorrect options:
(A) Kubelet does not run as a privileged container by default; it runs as a system process (typically systemd-managed) on the host.
(B) Kubelet does include built-in security features such asTLS authentication, authorization modes, and read-only vs secured ports.
(D) While kubelet interacts with the host system (e.g., cgroups, container runtimes), it does not inherently require root access for communication security; RBAC and TLS handle authentication.
[References:, Kubernetes Documentation – Kubelet authentication/authorization, CNCF Security Whitepaper – Cluster Component Security (discusses TLS and mutual authentication between kubelet and API server).,]

NEW QUESTION 51
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 56
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

Explanation:
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 57
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 62
.....

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