Online Linux Foundation KCNA Version & KCNA Study Center

To address the problems of Kubernetes and Cloud Native Associate (KCNA) exam candidates who are busy, VCEEngine has made the Kubernetes and Cloud Native Associate (KCNA) dumps PDF format of real Kubernetes and Cloud Native Associate (KCNA) exam questions. This format's feature to run on all smart devices saves your time. Because of this, the portability of Kubernetes and Cloud Native Associate (KCNA) dumps PDF aids in your preparation regardless of place and time restrictions.

Linux Foundation KCNA (Kubernetes and Cloud Native Associate) Exam is a certification program designed to test the proficiency of IT professionals in deploying and managing cloud-native applications using Kubernetes. Kubernetes is an open-source container orchestration platform that has become increasingly popular among organizations worldwide due to its ability to automate the deployment, scaling, and management of containerized applications. As a result, the demand for professionals with Kubernetes skills has been growing rapidly, making the KCNA Certification an important credential for those seeking career advancement in the cloud-native space.

>> Online Linux Foundation KCNA Version <<

Free PDF Quiz Linux Foundation - Trustable Online KCNA Version

It is time for you to plan your life carefully. After all, you have to make money by yourself. If you want to find a desirable job, you must rely on your ability to get the job. Now, our KCNA study materials will help you master the popular skills in the office. Believe it or not, our KCNA Study Materials will relieve you from poverty. It is important to make large amounts of money in modern society.

The KCNA exam is an online, performance-based exam that is conducted on a Kubernetes cluster. KCNA exam consists of a series of performance-based tasks that are designed to test the candidate's ability to deploy, manage, and troubleshoot Kubernetes clusters. KCNA Exam covers a wide range of topics, including container basics, Kubernetes architecture, deployment, networking, and storage.

Linux Foundation Kubernetes and Cloud Native Associate Sample Questions (Q25-Q30):

NEW QUESTION # 25
How do you deploy a workload to Kubernetes without additional tools?

- A. Create a Python script and run it with kubectl.
- B. Create a manifest and apply it with kubectl.
- C. Create a Bash script and run it on a worker node.
- D. Create a Helm Chart and install it with helm.

Answer: B

Explanation:

The standard way to deploy workloads to Kubernetes using only built-in tooling is to create Kubernetes manifests (YAML/JSON definitions of API objects) and apply them with kubectl, so C is correct. Kubernetes is a declarative system: you describe the desired state of resources (e.g., a Deployment, Service, ConfigMap, Ingress) in a manifest file, then submit that desired state to the API server. Controllers reconcile the actual cluster state to match what you declared.
A manifest typically includes mandatory fields like apiVersion, kind, and metadata, and then a spec describing desired behavior. For example, a Deployment manifest declares replicas and the Pod template (containers, images, ports, probes, resources). Applying the manifest with kubectl apply -f <file> creates or updates the resources. kubectl apply is also designed to work well with iterative changes: you update the file, re-apply, and Kubernetes performs a controlled rollout based on controller logic.
Option B (Helm) is indeed a popular deployment tool, but Helm is explicitly an "additional tool" beyond kubectl and the Kubernetes API. The question asks "without additional tools," so Helm is excluded by definition. Option A (running Bash scripts on worker nodes) bypasses Kubernetes' desired-state control and is not how Kubernetes workload deployment is intended; it also breaks portability and operational safety. Option D is not a standard Kubernetes deployment mechanism; kubectl does not "run Python scripts" to deploy workloads (though scripts can automate kubectl, that's still not the primary mechanism).
From a cloud native delivery standpoint, manifests support GitOps, reviewable changes, and repeatable deployments across environments. The Kubernetes-native approach is: declare resources in manifests and apply them to the cluster. Therefore, C is the verified correct answer.

NEW QUESTION # 26
Which prometheus metric type represents a single number value that can increase and decrease over time?

- A. Summary
- B. Gauge
- C. Histogram
- D. Counter

Answer: B

Explanation:
https://prometheus.io/docs/concepts/metric_types/#gauge

Gauge

A *gauge* is a metric that represents a single numerical value that can arbitrarily go up and down.

Gauges are typically used for measured values like temperatures or current memory usage, but also "counts" that can go up and down, like the number of concurrent requests.

NEW QUESTION # 27
What is the default service type in Kubernetes?

- A. serviceType
- B. loadBalancer
- C. NodePort
- D. CusterIP

Answer: D

Explanation:
https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types

ubernetes `ServiceTypes` allow you to specify what kind of Service ou want. The default is `ClusterIP`.

Type values and their behaviors are:

- `ClusterIP`: Exposes the Service on a cluster-internal IP. Choosing this value makes the Service only reachable from within the cluster. This is the default `ServiceType`.
- `NodePort`: Exposes the Service on each Node's IP at a static port (the `NodePort`). A `ClusterIP` Service, to which the `NodePort` Service routes, is automatically created. You'll be able to contact the `NodePort` Service, from outside the cluster, by requesting `<NodeIP>:<NodePort>`.
- `LoadBalancer`: Exposes the Service externally using a cloud provider's load balancer. `NodePort` and `ClusterIP` Services, to which the external load balancer routes, are automatically created.
- `ExternalName`: Maps the Service to the contents of the `externalName` field (e.g. `foo.bar.example.com`), by returning a `CNAME` record with its value. No proxying of any kind is set up.

NEW QUESTION # 28

Which statement about Secrets is correct?

- A. Secret data is encrypted with the cluster private key by default.
- B. A Secret is part of a Pod specification.
- C. Secret data is base64 encoded and stored unencrypted by default.
- D. A Secret can only be used for confidential data.

Answer: C

Explanation:
The correct answer is C. By default, Kubernetes Secrets store their data as base64-encoded values in the API (backed by etcd). Base64 is an encoding mechanism, not encryption, so this does not provide confidentiality. Unless you explicitly configure encryption at rest for etcd (via the API server encryption provider configuration) and secure access controls, Secret contents should be treated as potentially readable by anyone with sufficient API access or access to etcd backups.
Option A is misleading: a Secret is its own Kubernetes resource (kind: Secret). While Pods can reference Secrets (as environment variables or mounted volumes), the Secret itself is not "part of the Pod spec" as an embedded object. Option B is incorrect because Kubernetes does not automatically encrypt Secret data with a cluster private key by default; encryption at rest is optional and must be enabled. Option D is incorrect because Secrets can store a range of sensitive or semi-sensitive data (tokens, certs, passwords), but Kubernetes does not enforce "only confidential data" semantics; it's a storage mechanism with size and format constraints.
Operationally, best practices include: enabling encryption at rest, limiting access via RBAC, avoiding broad "list/get secrets" permissions, using dedicated service accounts, auditing access, and considering external secrets managers (Vault, cloud KMS-backed solutions) for higher assurance. Also, don't confuse "Secret" with "secure by default." The default protection is mainly about avoiding accidental plaintext exposure in manifests, not about cryptographic security.
So the only correct statement in the options is C.

NEW QUESTION # 29

Observability and monitoring are not the same?

- A. False
- B. True

Answer: B

NEW QUESTION # 30

......

KCNA Study Center: https://www.vceengine.com/KCNA-vce-test-engine.html