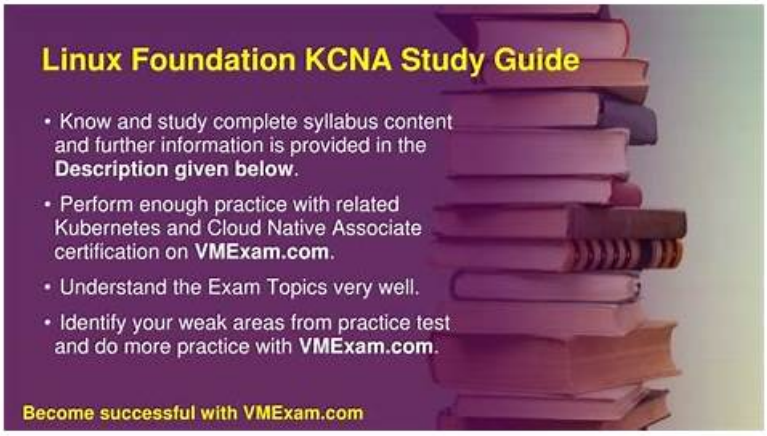
# Formats of Linux Foundation KCNA Practice Exam Questions




P.S. Free & New KCNA dumps are available on Google Drive shared by Actual4Labs: https://drive.google.com/open?id=10l2U0GUF7nrQeZ-ioHAvNRUcABGSJ3HE

We've always put quality of our KCNA study guide on top priority. We don't strongly chase for the number of products we have manufactured. Each KCNA test engine will go through strict inspection from many aspects such as the operation, compatibility test and so on. Also, we have final random sampling survey before we sale our KCNA practice material to our customers. The quality inspection process is completely strict. The most professional experts of our company will check the study guide and deal with the wrong parts. What you have bought will totally have no problem. That is why we can survive in the market now. Our company is dedicated to carrying out the best quality KCNA Test Engine. Any small mistake is intolerant. You can buy our products at ease.

Linux Foundation KCNA exam is an online, proctored exam that can be taken from anywhere in the world. KCNA exam consists of 40 multiple-choice questions and must be completed within 90 minutes. KCNA Exam is designed to be challenging, but fair, and is intended to test the candidate's knowledge and understanding of Kubernetes and cloud-native technologies.

**>> KCNA Study Demo <<**

## 2026 Linux Foundation KCNA: Kubernetes and Cloud Native Associate – Valid Study Demo

We also offer up to 365 days free KCNA exam dumps updates. These free updates will help you study as per the KCNA latest examination content. Our valued customers can also download a free demo of our Kubernetes and Cloud Native Associate KCNA Exam Dumps before purchasing. We guarantee 100% satisfaction for our KCNA practice material users, thus our Kubernetes and Cloud Native Associate KCNA study material saves your time and money.

## Linux Foundation Kubernetes and Cloud Native Associate Sample Questions (Q33-Q38):

**NEW QUESTION # 33**
Let's assume that an organization needs to process large amounts of data in bursts, on a cloud-based Kubernetes cluster. For instance: each Monday morning, they need to run a batch of 1000 compute jobs of 1 hour each, and these jobs must be completed by Monday night. What's going to be the most cost-effective method?

- A. Leverage the Kubernetes Cluster Autoscaler to automatically start and stop nodes as they're needed.
- B. Run a group of nodes with the exact required size to complete the batch on time, and use a combination of taints, tolerations, and nodeSelectors to reserve these nodes to the batch jobs.
- C. Commit to a specific level of spending to get discounted prices (with e.g. "reserved instances" or similar mechanisms).
- D. Use PriorityClasses so that the weekly batch job gets priority over other workloads running on the cluster, and can be completed on time.

**Answer: A**

Explanation:
Burst workloads are a classic elasticity problem: you need large capacity for a short window, then very little capacity the rest of the week. The most cost-effective approach in a cloud-based Kubernetes environment is to scale infrastructure dynamically, matching node count to current demand. That's exactly what Cluster Autoscaler is designed for: it adds nodes when Pods cannot be scheduled due to insufficient resources and removes nodes when they become underutilized and can be drained safely. Therefore B is correct.
Option A can work operationally, but it commonly results in paying for a reserved "standing army" of nodes that sit idle most of the week-wasteful for bursty patterns unless the nodes are repurposed for other workloads. Taints/tolerations and nodeSelectors are placement tools; they don't reduce cost by themselves and may increase waste if they isolate nodes. Option D (PriorityClasses) affects which Pods get scheduled first given available capacity, but it does not create capacity. If the cluster doesn't have enough nodes, high priority Pods will still remain Pending. Option C (reserved instances or committed-use discounts) can reduce unit price, but it assumes relatively predictable baseline usage. For true bursts, you usually want a smaller baseline plus autoscaling, and optionally combine it with discounted capacity types if your cloud supports them.
In Kubernetes terms, the control loop is: batch Jobs create Pods → scheduler tries to place Pods → if many Pods are Pending due to insufficient CPU/memory, Cluster Autoscaler observes this and increases the node group size → new nodes join and kube-scheduler places Pods → after jobs finish and nodes become empty, Cluster Autoscaler drains and removes nodes. This matches cloud-native principles: elasticity, pay-for-what-you-use, and automation. It minimizes idle capacity while still meeting the completion deadline.

**NEW QUESTION # 34**
You have a Kubernetes cluster with multiple namespaces. You are running different applications in separate namespaces and want to ensure that HPAs in each namespace are not interfering with each other. Which of the following measures should you take?

- A. Use a single HPA for all applications in all namespaces.
- B. There is no need for any specific measures as HPAs are isolated by default within their respective namespaces.
- C. Ensure that the •minReplicas• and •maxReplicaS values for HPAs across namespaces are aligned.
- D. Create a custom controller to manage HPAs across multiple namespaces and coordinate scaling decisions.
- E. Configure resource quotas for each namespace to prevent resource over-allocation by HPAs.

**Answer: D,E**

Explanation:
You need to ensure that HPAs are not competing for resources across namespaces- Resource quotas (C) limit resource consumption within each namespace, preventing HPAs from consuming resources intended for other applications. A custom controller (D) can be used to manage HPAs across multiple namespaces, coordinating scaling decisions to avoid resource contention and optimize utilization across the cluster. While options A and B might seem relevant, they are not effective solutions for preventing interference between HPAs in different namespaces. Option E is incorrect; HPAs operate within their namespaces, but resource contention can occur if not managed properly.

**NEW QUESTION # 35**
Which style of operations are preferred for K8S and cloud native applications?

- A. JSON
- B. Imperative
- C. Declarative

**Answer: C**

Explanation:
https://kubernetes.io/docs/tasks/manage-kubernetes-objects/declarative-config/#trade-offs

**NEW QUESTION # 36**
You have a Kubernetes cluster with two worker nodes. One node has 8 CPU cores and 16GB RAM, while the other has 4 CPU cores and 8GB RAM. You deploy a pod with resource requests of 2 CPU cores and 4GB RAM. Where is this pod most likely to be scheduled?

- A. The node with 4 CPU cores and 8GB RAM.

- B. The node with the highest available memory capacity.
- C. The node with the highest available CPU capacity.
- D. The node with 8 CPU cores and 16GB RAM.
- E. Either node, as Kubernetes will randomly choose.

**Answer: D**

Explanation:
Kubernetes will try to schedule pods on nodes that have enough resources to meet the pod's requests. In this case, both nodes have enough resources, but the node with 8 CPU cores and 16GB RAM has more available resources, making it the more likely candidate for the pod to be scheduled on.

**NEW QUESTION # 37**
You're developing a new microservice for an application running on Kubernetes. The service requires access to a database hosted in a separate Kubernetes cluster. How would you ensure secure communication between your microservice and the database?

- A. Configure a Kubernetes Ingress resource to route traffic to the database
- B. Use a shared secret stored in an environment variable within the microservice pod
- C. Use an API gateway to proxy requests between the services.
- D. Configure a service mesh to handle encryption and authentication between the services.
- E. Set up a VPN connection between the two Kubernetes clusters-

**Answer: D**

Explanation:
A service mesh like Istio or Linkerd provides a layer of abstraction that can handle encryption, authentication, and authorization bewveen microservices within and across Kubernetes clusters. It allows for secure communication without exposing sensitive credentials directly within the microservice code.

**NEW QUESTION # 38**
.....

- KCNA Practice Exam Fee ～ KCNA Practice Exam Fee □ KCNA Latest Exam Fee □ Search for □ KCNA □ and easily obtain a free download on ✔ www.verifieddumps.com □✔□ □100% KCNA Correct Answers
- www.stes.tyc.edu.tw, www.lynxnlearn.com, dl.instructure.com, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, codematetv.com, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, elearning.eauqardho.edu.so, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, Disposable vapes

P.S. Free 2026 Linux Foundation KCNA dumps are available on Google Drive shared by Actual4Labs: https://drive.google.com/open?id=10l2U0GUF7nrQeZ-ioHAvNRUcABGSJ3HE