

# Well KCNA Prep & New KCNA Study Plan



P.S. Free 2026 Linux Foundation KCNA dumps are available on Google Drive shared by TopExamCollection:  
<https://drive.google.com/open?id=1QeXf9VyTxbzIYUG4Rsnnu-mpkHmSkDHb>

We will give you free update for 365 days after purchasing KCNA study guide from us, that is to say, in the following year, you don't need to spend extra money on update version, and the latest version for KCNA exam dumps will be sent to your email address automatically. Furthermore, KCNA exam dumps are high quality and accuracy, and they can help you pass the exam just one time. In order to strengthen your confidence to KCNA Study Guide, we are pass guarantee and money back guarantee, if you fail to pass the exam we will give you full refund, and there is no need for you to worry about that you will waste your money.

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.

>> Well KCNA Prep <<

## HOT Well KCNA Prep - High-quality Linux Foundation New KCNA Study Plan: Kubernetes and Cloud Native Associate

This allows candidates to choose the format that best suits their learning style and preference, ensuring a seamless and effective exam preparation experience. By offering tailored solutions to meet individual needs, TopExamCollection has established itself as a trusted provider of top-quality Kubernetes and Cloud Native Associate (KCNA) exam preparation material.

## Linux Foundation Kubernetes and Cloud Native Associate Sample Questions (Q137-Q142):

### NEW QUESTION # 137

Continuous delivery is \_\_\_\_\_.

- A. Coding, Building and Testing the code
- **B. Automatically deploying code to [container or server] environment**
- C. Manually deploying the code

**Answer: B**

### NEW QUESTION # 138

The three typical opentelemetry data is?

- A. Metrics
- **B. All of the options**
- C. Logs
- D. Traces

**Answer: B**

Explanation:

<https://opentelemetry.io/docs/concepts/data-sources/>

#### NEW QUESTION # 139

How does Horizontal Pod autoscaling work in Kubernetes?

- A. The Horizontal Pod Autoscaler controller adds more pods when the load is above the configured threshold, and reduces the number of pods when the load is below.
- B. The Horizontal Pod Autoscaler controller adds more pods when the load is above the configured threshold, but does not reduce the number of pods when the load is below.
- C. The Horizontal Pod Autoscaler controller adds more pods to the specified DaemonSet when the load is above the configured threshold, and reduces the number of pods when the load is below.
- D. The Horizontal Pod Autoscaler controller adds more CPU or memory to the pods when the load is above the configured threshold, and reduces CPU or memory when the load is below.

**Answer: A**

Explanation:

Horizontal Pod Autoscaling (HPA) adjusts the number of Pod replicas for a workload controller (most commonly a Deployment) based on observed metrics, increasing replicas when load is high and decreasing when load drops. That matches D, so D is correct. HPA does not add CPU or memory to existing Pods—that would be vertical scaling (VPA). Instead, HPA changes spec.replicas on the target resource, and the controller then creates or removes Pods accordingly.

HPA commonly scales based on CPU utilization and memory (resource metrics), and it can also scale using custom or external metrics if those are exposed via the appropriate Kubernetes metrics APIs.

Option A is vertical scaling behavior, not HPA. Option B is incorrect because HPA can scale down as well as up (subject to stabilization windows and configuration), so it's not "scale up only." Option C is incorrect because HPA does not scale DaemonSets in the usual model; DaemonSets are designed to run one Pod per node (or per selected nodes) rather than a replica count. HPA targets resources like Deployments, ReplicaSets (via Deployment), and StatefulSets in typical usage, where replica count is a meaningful knob.

Operationally, HPA works as a control loop: it periodically reads metrics (for example, via metrics-server for CPU/memory, or via adapters for custom metrics), compares the current value to the desired target, and calculates a desired replica count within min/max bounds. To avoid flapping, HPA includes stabilization behavior and cooldown logic so it doesn't scale too aggressively in response to short spikes or dips. You can configure minimum and maximum replicas and behavior policies to tune responsiveness.

In cloud-native systems, HPA is a key elasticity mechanism: it enables services to handle variable traffic while controlling cost by scaling down during low demand. Therefore, the verified correct answer is D.

#### NEW QUESTION # 140

What is the difference between a Service and an Ingress in Kubernetes?

- A. Services manage the lifecycle of Pods, while Ingress manages the lifecycle of Deployments.
- B. Services expose applications running within the cluster, while Ingress provides routing and load balancing for external traffic.
- C. Services are used for scheduling Pods to nodes, while Ingress is used for managing the communication between Pods and services.
- D. Services are used for managing the state of Pods, while Ingress is used for managing the health of Pods.
- E. Services provide external access to your application, while Ingress provides internal access.

**Answer: B**

Explanation:

In Kubernetes Services provide a way to expose applications running within the cluster, while Ingress provides a mechanism for routing and load balancing external traffic to your applications. Services are used to make Pods accessible within the cluster, while Ingress enables users outside the cluster to access your applications.

### NEW QUESTION # 141

Your application is deployed in a Kubernetes cluster and is experiencing performance issues. You need to monitor the health of the cluster and identify potential bottlenecks. Which Kubernetes features would be most helpful for diagnosing and resolving these issues?

- A. Jaeger for tracing distributed requests
- B. Prometheus and Grafana for metrics collection and visualization
- C. Kubernetes Dashboard for visual overview of cluster status
- **D. All of the above**
- E. kubectl logs and kubectl describe

**Answer: D**

Explanation:

All of the mentioned features contribute to effective cluster monitoring and debugging: 'kubectl logs' and 'kubectl describe' provide valuable information about specific Pods and their resources. Prometheus and Grafana allow you to collect and analyze metrics from various components of the cluster, helping you identify performance bottlenecks. Kubernetes Dashboard offers a visual overview of the cluster's state and resources. Jaeger is a powerful tool for tracing distributed requests across different microservices within your application, helping you diagnose issues that involve communication between services.

**NEW QUESTION # 142**

• • • • •

As the old saying tells that, he who doesn't go advance will lose his ground. So you will have a positive outlook on life. All in all, abandon all illusions and face up to reality bravely. Our KCNA practice exam will be your best assistant to get the KCNA Certification. And our KCNA study materials are always considered the guarantee to pass the exam. You are the best and unique in the world. Just be confident to face new challenge!

**New KCNA Study Plan:** <https://www.topexamcollection.com/KCNA-vce-collection.html>

- [illegible]

What's more, part of that TopExamCollection KCNA dumps now are free: <https://drive.google.com/open?id=1QeXt9VyTxbzIYUG4Rsnnu-mpkHmSkDHb>