KUBERNETES CERTIFIED NETWORK ADMINISTRATOR (KCNA) PART 2 PRACTICE TEST (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR THE FULL VERSION STUDY GUIDE

https://kcnapt2.examzify.com

Copyright © 2026 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

Notice: Examzify makes every reasonable effort to obtain accurate, complete, and timely information about this product from reliable sources.

SAMPLE

Table of Contents

Copyright 1

Table of Contents 2

Introduction 3

How to Use This Guide 4

Questions 5

Answers 8

Explanations 10

Next Steps 15

SAMPLE

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

SAMPLE

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

SAMPLE

Questions

SAMPLE

1. **Which Linux feature is used by Kubernetes to isolate containers and limit the resource usage of a process or set of processes?**

   A. Namespacess

   B. Cgroups

   C. Chroot

   D. AppArmor

2. **Which statement best describes the RollingUpdate strategy in Kubernetes Deployments?**

   A. To incrementally update pods with a new version while maintaining availability.

   B. To replace all pods at once.

   C. To scale pods down to zero.

   D. To delete the deployment.

3. **What is the primary purpose of KubeScape in Kubernetes security?**

   A. Monitor cluster performance metrics

   B. Assess security posture of Kubernetes clusters

   C. Automate workload deployment

   D. Scan container images for vulnerabilities

4. **Which Kubernetes resources support immutable: true to prevent changes after creation?**

   A. Deployments and StatefulSets

   B. Pods and Nodes

   C. ConfigMaps and Secrets

   D. Services and Ingresses

5. **What is the general behavior of Kubernetes Deployment's default update strategy during updates?**

   A. Update all pods simultaneously

   B. Permanently stop updates after first failure

   C. Only update environment variables

   D. Update pods one by one to minimize downtime

SAMPLE

6. Which Kubernetes object is most suitable for deploying stateless applications?

A. Deployment

B. StatefulSet

C. DaemonSet

D. Job

7. Which tool is specifically designed for monitoring and managing costs in a Kubernetes environment, providing insights into spending and resource optimisation?

A. Prometheus

B. Grafana

C. Kubecost

D. Kubernetes Dashboard

8. Function associated with Kubernetes Security Contexts.

A. Scheduling containers on nodes

B. Defining container or pod level security settings

C. Managing network policies

D. Enforcing PodSecurityPolicies

9. In a managed Kubernetes service, which component handles cloud provider integration for resources like LoadBalancer?

A. Cluster Autoscaler

B. Cloud Controller Manager

C. Kube Proxy

D. CoreDNS

10. Which Kubernetes service type is NOT considered a standard service type?

A. ClusterIP

B. NodePort

C. Ingress

D. LoadBalancer

SAMPLE

Answers

SAMPLE

1. B
2. A
3. B
4. C
5. D
6. A
7. C
8. B
9. B
10. C

SAMPLE

Explanations

SAMPLE

1. Which Linux feature is used by Kubernetes to isolate containers and limit the resource usage of a process or set of processes?

A. Namespacess

B. Cgroups

C. Chroot

D. AppArmor

Cgroups provide the mechanism for both isolating and enforcing resource limits for containers. In Kubernetes, each container runs inside its own control group, and the Pod spec's resource requests and limits are translated into CPU, memory, and other quota boundaries applied to that group. This ensures a container can't exceed its allocated resources and helps maintain node stability by preventing one container from starving others. Namespaces handle isolation of kernel resources like PID trees, network stacks, and mounts, but they don't enforce resource usage limits. Chroot is a filesystem isolation technique, and AppArmor offers security profiles rather than resource quotas. Therefore, the feature that Kubernetes uses to isolate containers and cap their resource usage is cgroups.

2. Which statement best describes the RollingUpdate strategy in Kubernetes Deployments?

A. To incrementally update pods with a new version while maintaining availability.

B. To replace all pods at once.

C. To scale pods down to zero.

D. To delete the deployment.

RollingUpdate updates pods gradually, replacing old pods with new ones one by one while keeping the service available throughout the process. Kubernetes accomplishes this by creating new pods with the updated version, waiting for them to become ready, and then terminating old pods, repeating until all are updated. This approach preserves availability because a portion of the old pods remains serving traffic while the new pods come online. The behavior is controlled by settings like maxUnavailable (how many pods can be down during the update) and maxSurge (how many extra pods can run temporarily). For example, with a deployment of five replicas and default RollingUpdate settings, you might briefly run six pods during the update and never have fewer than four available pods, ensuring continuous service. This is in contrast to replacing all pods at once, scaling down to zero, or deleting the deployment, which would disrupt service or remove the workload management.

3. What is the primary purpose of KubeScape in Kubernetes security?

A. Monitor cluster performance metrics

B. Assess security posture of Kubernetes clusters

C. Automate workload deployment

D. Scan container images for vulnerabilities

KubeScape is about measuring and improving your security posture in Kubernetes. Its primary purpose is to continuously assess how securely your clusters are configured by evaluating them against security policies and best practices. It identifies risk indicators such as overly permissive RBAC, misconfigured network policies, exposed secrets, insecure defaults, and other configuration issues that could be exploited. By producing a prioritized risk view and actionable remediation steps, it helps operators harden the cluster and reduce attack surface. This differs from monitoring cluster performance metrics, which focuses on resource usage and health signals; automating workload deployment, which handles delivery of applications; or scanning container images for vulnerabilities, which targets issues inside container images rather than cluster configuration and policy posture.

SAMPLE

4. Which Kubernetes resources support immutable: true to prevent changes after creation?

A. Deployments and StatefulSets

B. Pods and Nodes

C. ConfigMaps and Secrets

D. Services and Ingresses

Immutability in Kubernetes is about preventing in-place changes to certain objects after they're created. For a resource to be immutable, you can set a flag (immutable: true) on it when you create it. Once that flag is set, updating the data of that object in place is not allowed; changes must be done by deleting and recreating the object (often with a new name) and then updating any references to point to the new one. ConfigMaps and Secrets are the resources that support immutable: true. This makes their data fixed after creation, which is particularly useful for protecting configuration data and sensitive information from accidental or unintended changes. If you need to update them, you create a new ConfigMap or Secret (often with a new name or version) and update the workload to use the new one. Other resource types like Deployments and StatefulSets, Pods, Nodes, Services, and Ingresses are designed to be updated and managed over time through changes to their specs or endpoints; they do not offer an immutable flag to lock their data in place.

5. What is the general behavior of Kubernetes Deployment's default update strategy during updates?

A. Update all pods simultaneously

B. Permanently stop updates after first failure

C. Only update environment variables

D. Update pods one by one to minimize downtime

Rolling updates are the default approach for Kubernetes Deployments. Instead of updating all pods at once, the system replaces old pods with new ones in small steps, gradually rolling through the set of replicas. This keeps the application available throughout the upgrade, because some pods remain serving traffic while others are updated. The exact pace is influenced by settings like maxUnavailable and maxSurge, but the core idea is that updates happen in a rolling manner to minimize downtime. So updating pods one by one (or in small batches) to maintain service availability is the best description.

6. Which Kubernetes object is most suitable for deploying stateless applications?

A. Deployment

B. StatefulSet

C. DaemonSet

D. Job

Deployments are the go-to way to run stateless applications in Kubernetes. They manage a desired number of replicas through a controlling ReplicaSet, handle rolling updates and rollbacks, and keep pods interchangeable. Because stateless pods don't rely on stable identity or local storage, you don't need to preserve a specific pod name or startup order; a Deployment provides a reliable, declarative way to scale, update, and recover your service while keeping a stable service endpoint via a separate Kubernetes Service. The other options serve different needs. StatefulSet is for stateful workloads that require stable network identities and persistent storage. DaemonSet runs a copy of a pod on every node (or a selected subset), which is great for node-level agents but not for general stateless services. Job is for batch tasks that run to completion and finish, not for long-running stateless services.

SAMPLE

7. Which tool is specifically designed for monitoring and managing costs in a Kubernetes environment, providing insights into spending and resource optimisation?

A. Prometheus

B. Grafana

C. Kubecost

D. Kubernetes Dashboard

Cost visibility and optimization in Kubernetes is about tying resource usage to actual spend and making informed decisions to reduce waste. Kubecost is built for this purpose: it provides cost allocation and optimization insights across your cluster, broken down by namespace, deployment, label, service, or workload, and it can show you spend across cloud providers and clusters. Kubecost pulls together cloud billing data with Kubernetes usage to give real-time and historical views of cost, plus budgets, alerts, and forecasting. It helps you spot waste—like over-provisioned or idle resources—and offers practical optimization guidance, such as right-sizing resources or adjusting schedules. While Prometheus collects metrics, Grafana visualizes them, and the Kubernetes Dashboard offers cluster management UI, none of these focus on cost accounting and optimization. Kubecost specifically targets that cost perspective, making it the best fit for monitoring and managing Kubernetes expenses.

8. Function associated with Kubernetes Security Contexts.

A. Scheduling containers on nodes

B. Defining container or pod level security settings

C. Managing network policies

D. Enforcing PodSecurityPolicies

Security contexts specify the security attributes applied to a container or pod, such as which user the process runs as, whether the root filesystem is read-only, and which capabilities or SELinux/AppArmor labels are allowed or dropped. These settings define the runtime security posture and enforce least-privilege execution for containers, shaping how they operate inside the pod. Scheduling containers on nodes is handled by the scheduler using node selectors, taints, tolerations, and affinity rules, not by security contexts. Managing network policies controls which pods can talk to which—it's about network traffic, not container runtime security settings. Enforcing PodSecurityPolicies (or Pod Security Standards) governs overarching policy constraints on pods, but security contexts are the specific per-container or per-pod security attributes you configure inside the Pod spec to control privileges and capabilities.

9. In a managed Kubernetes service, which component handles cloud provider integration for resources like LoadBalancer?

A. Cluster Autoscaler

B. Cloud Controller Manager

C. Kube Proxy

D. CoreDNS

The Cloud Controller Manager handles cloud provider integration for resources like LoadBalancer. It runs cloud-specific controllers that talk to the cloud provider's API to provision and manage external resources such as load balancers, routes, and other cloud resources, keeping the Kubernetes control plane itself cloud-agnostic. When you create a Service of type LoadBalancer, the Cloud Controller Manager provisions the external load balancer in the cloud and connects it to the service. Other components serve different roles: Kube Proxy manages in-cluster networking for services, CoreDNS provides DNS for service discovery, and the Cluster Autoscaler adjusts the number of nodes.

SAMPLE

10. Which Kubernetes service type is NOT considered a standard service type?

A. ClusterIP

B. NodePort

C. Ingress

D. LoadBalancer

Exposing applications in Kubernetes uses Service resources that come in a few standard types: ClusterIP for internal access, NodePort to expose on node ports, and LoadBalancer to provision an external load balancer. Ingress is not a Service type; it's a separate resource that defines HTTP/HTTPS routing rules and relies on an Ingress Controller to implement those rules. With Ingress you can route traffic to multiple backend services behind a single IP using hostnames and paths, which is a different mechanism from a Service type that directly exposes endpoints for one service. So, the option that isn't a standard service type is Ingress. In practice, you typically have internal Services (often ClusterIP) and an Ingress resource with a controller to provide external access and traffic routing to those services.

SAMPLE

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

https://kcnapt2.examzify.com

We wish you the very best on your exam journey. You've got this!

SAMPLE