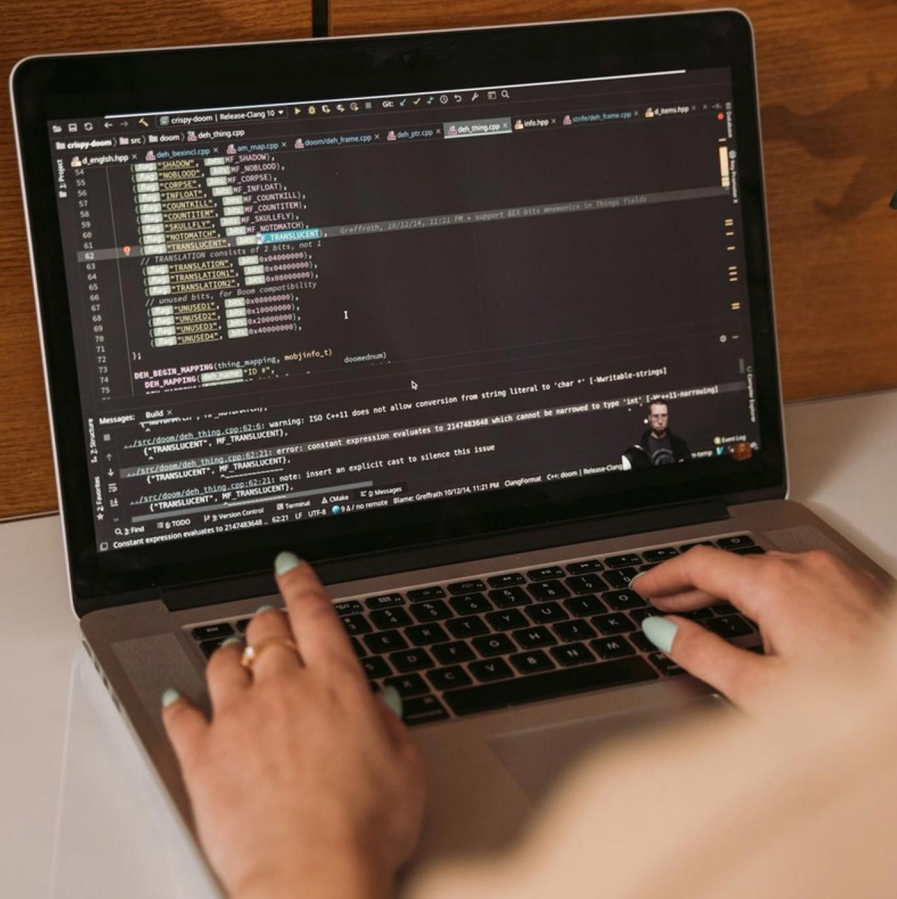


RED HAT OPENSIFT DEVELOPER EX288 PRACTICE EXAM (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR
THE FULL VERSION STUDY GUIDE

<https://redhatopenshiftdevex288.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	9
Explanations	11
Next Steps	16

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.

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!

Questions

SAMPLE

- 1. In Kubernetes/OpenShift, which statement correctly describes Pod and Deployment?**
 - A. Deployment is the smallest unit; a Pod manages nodes and networking.
 - B. A Pod is the smallest unit; a Deployment manages a set of Pods and handles updates and scaling.
 - C. Pods and Deployments are equivalent concepts used interchangeably.
 - D. A Pod runs a single container only; a Deployment runs multiple containers.

- 2. In kustomize, where do you define the files to be built for a customization?**
 - A. in kustomization.yaml
 - B. in deployment.yaml and secrets.yaml
 - C. under metadata
 - D. with apiVersion field

- 3. Which resource references a StorageClass to provision storage for a pod, and what does the StorageClass define?**
 - A. Deployment references a StorageClass; StorageClass defines provisioning parameters and storage type.
 - B. PersistentVolumeClaim references a StorageClass; StorageClass defines access modes.
 - C. PersistentVolume references a StorageClass; StorageClass defines retention policy.
 - D. Job references a StorageClass; StorageClass defines provisioning parameters.

- 4. Which statement best describes NetworkPolicy in OpenShift and how it affects pod traffic?**
 - A. It defines allowed ingress/egress rules; by default all traffic is allowed; with policies you can restrict traffic; requires cluster support.
 - B. It defines quotas for storage per namespace.
 - C. It creates Service accounts automatically.
 - D. It manages encryption for Secrets.

5. What is an ImageStream in OpenShift and what is its purpose?

- A. ImageStreams track images and help trigger deployments when a new image is available.
- B. ImageStreams store logs for auditing.
- C. ImageStreams define network policies for pods.
- D. ImageStreams manage Secrets for applications.

6. What is the purpose of a Route in OpenShift?

- A. To configure internal network policies.
- B. To monitor pod metrics.
- C. To expose an application to external clients by mapping a DNS name to a service.
- D. To manage image tags.

7. What is the recommended directory structure for an application named myapp in a kustomize project?

- A. myapp/base and myapp/overlays with production, staging, and dev under overlays
- B. myapp/config and myapp/overlays with no base
- C. myapp/root and myapp/env
- D. src/myapp and assets

8. Which statement about values.yaml is true?

- A. It defines the chart's metadata and dependencies.
- B. It lists the resources to deploy.
- C. It stores release history.
- D. It holds configuration values that can be overridden.

9. Which object controls the number of pod replicas for an application in OpenShift?

- A. Service
- B. ConfigMap
- C. Deployment
- D. Route

10. What is the command to create a configmap cm3 with the literal value DB_HOST=localhost?

- A. `oc create cm3 --from-literal DB_HOST=localhost`
- B. `oc create configmap cm3 --from-file /tmp/values.txt`
- C. `oc create configmap cm3 --from-literal DB_HOST=localhost`
- D. `oc create configmap cm3 --from-file /etc/db_host.txt`

SAMPLE

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. In Kubernetes/OpenShift, which statement correctly describes Pod and Deployment?

- A. Deployment is the smallest unit; a Pod manages nodes and networking.
- B. A Pod is the smallest unit; a Deployment manages a set of Pods and handles updates and scaling.**
- C. Pods and Deployments are equivalent concepts used interchangeably.
- D. A Pod runs a single container only; a Deployment runs multiple containers.

In Kubernetes and OpenShift, the Pod is the smallest schedulable unit that runs one or more containers in a shared context (network and storage). A Deployment sits above Pods as a controller that defines how many Pods should run and how updates and scaling should be handled. It creates and manages ReplicaSets, which in turn ensure the desired number of Pod replicas, enabling safe rolling updates and automatic rollbacks if something goes wrong. Pods can host multiple containers, though common practice is one container per Pod; a Deployment doesn't run containers directly, it manages the Pods that do. This is why selecting the statement that a Pod is the smallest unit and a Deployment manages a set of Pods with updates and scaling is the best description.

2. In kustomize, where do you define the files to be built for a customization?

- A. in kustomization.yaml**
- B. in deployment.yaml and secrets.yaml
- C. under metadata
- D. with apiVersion field

Kustomize builds are defined in a kustomization.yaml in your directory. This file declares which files to process by listing them under the resources section, along with any patches and other customization steps. The actual manifest files (such as deployment.yaml or secrets.yaml) live alongside, but kustomize uses the list in kustomization.yaml to know which files to build and transform. Fields like metadata or apiVersion belong to the manifests themselves and aren't where you specify the set of files to include. So the place to define the files to be built for a customization is the kustomization.yaml.

3. Which resource references a StorageClass to provision storage for a pod, and what does the StorageClass define?

- A. Deployment references a StorageClass; StorageClass defines provisioning parameters and storage type.
- B. PersistentVolumeClaim references a StorageClass; StorageClass defines access modes.**
- C. PersistentVolume references a StorageClass; StorageClass defines retention policy.
- D. Job references a StorageClass; StorageClass defines provisioning parameters.

The key idea is dynamic storage provisioning. A Pod doesn't talk to a StorageClass directly; instead, a PersistentVolumeClaim asks for storage and can specify a StorageClass by name. When the PVC is created with that StorageClass, the cluster's provisioner creates a matching PersistentVolume to satisfy the claim, and the Pod can then mount it via the PVC. The StorageClass defines how to provision storage: which provisioner to use (the driver that actually creates the volume) and the provisioning parameters (such as storage type, performance characteristics, and other driver-specific options). It can also set handling details like reclaimPolicy and volumeBindingMode. Access modes are properties of the PVC and the resulting PV, not of the StorageClass itself. So, the resource that references a StorageClass to provision storage for a pod is the PersistentVolumeClaim, and the StorageClass describes the provisioning parameters and the storage type.

4. Which statement best describes NetworkPolicy in OpenShift and how it affects pod traffic?

- A. It defines allowed ingress/egress rules; by default all traffic is allowed; with policies you can restrict traffic; requires cluster support.**
- B. It defines quotas for storage per namespace.
- C. It creates Service accounts automatically.
- D. It manages encryption for Secrets.

NetworkPolicy is about controlling how pods talk to each other by defining allowed ingress and egress rules. In OpenShift, the enforcement of these rules depends on the cluster's network plugin (the CNI) supporting NetworkPolicy. If no policy selects a pod, that pod can communicate freely with others. When a policy does select certain pods, traffic to and from those pods is restricted to what the policy permits, effectively isolating them unless explicit allowances exist. This is how you implement precise control over pod communication and segment traffic. The other statements don't describe traffic control at all: quotas for storage, automatic creation of service accounts, and encryption management for Secrets are unrelated to how NetworkPolicy governs pod traffic.

5. What is an ImageStream in OpenShift and what is its purpose?

- A. ImageStreams track images and help trigger deployments when a new image is available.**
- B. ImageStreams store logs for auditing.
- C. ImageStreams define network policies for pods.
- D. ImageStreams manage Secrets for applications.

In OpenShift, an ImageStream is a resource that tracks container images within a project and provides a stable reference to a specific image tag in the cluster. Its purpose is to surface updates to images in a decoupled way so deployments can react automatically when a new image is available. When you push a new image and update the ImageStreamTag, OpenShift can trigger a new deployment for your application if an image change trigger is configured, allowing seamless automated rollouts without hard-coding image details in pod specs. This makes workflows like continuous delivery easier, as deployments can be driven by image updates to the ImageStream rather than manual edits. It's not used for storing logs, defining network policies, or managing Secrets; those are handled by other resources and components in OpenShift.

6. What is the purpose of a Route in OpenShift?

- A. To configure internal network policies.
- B. To monitor pod metrics.
- C. To expose an application to external clients by mapping a DNS name to a service.**
- D. To manage image tags.

In OpenShift, a Route is used to expose an application to external clients by mapping a DNS name to a service. The router watches Routes and configures the underlying load balancer to direct external traffic that matches the host (and optional path) to the Service's pods, enabling access from outside the cluster. This setup often allows TLS termination at the router and convenient DNS-based access to your app. It's not about internal network policies, pod metrics, or image tag management, which are separate functions. So the Route's purpose is to provide external exposure of an app by linking a DNS name to a Service.

7. What is the recommended directory structure for an application named myapp in a kustomize project?

- A. myapp/base and myapp/overlays with production, staging, and dev under overlays**
- B. myapp/config and myapp/overlays with no base
- C. myapp/root and myapp/env
- D. src/myapp and assets

In a kustomize project, you keep the common resources that apply to all environments in a base, and you create environment-specific modifications in overlays. The base holds the shared manifests (Deployment, Service, ConfigMap, etc.), while each overlay (for production, staging, dev) applies patches or its own config on top of that base. This approach lets you reuse the same core definitions and only change what's different per environment, making deployments consistent and easier to manage. So for an app named myapp, the recommended structure is to have myapp/base for the common manifests and myapp/overlays with subfolders like production, staging, and dev that reference the base and provide the environment-specific tweaks. The other options lack a base, use nonstandard layouts, or place resources in unrelated folders, which undermines reuse and clean environment separation.

8. Which statement about values.yaml is true?

- A. It defines the chart's metadata and dependencies.
- B. It lists the resources to deploy.
- C. It stores release history.
- D. It holds configuration values that can be overridden.**

In Helm charts, values.yaml serves as the configuration input for templates. It provides default values for things like replica counts, image tags, resource requests, and feature flags. These values are injected into the chart's templates through the .Values object, so the actual Kubernetes manifests can be rendered with these settings. The key point is that these defaults can be overridden when you install or upgrade the chart. You can supply your own values file or use --set to override specific keys, allowing customization without changing the templates themselves. This makes values.yaml the mechanism for flexible configuration across different environments. The other statements point to different parts of a chart: metadata and dependencies are defined in Chart.yaml, the resources to deploy come from the templates, and release history is tracked by Helm's release storage rather than by values.yaml.

9. Which object controls the number of pod replicas for an application in OpenShift?

- A. Service
- B. ConfigMap
- C. Deployment**
- D. Route

In OpenShift, the Deployment is what controls how many pods run for an application. It defines the desired number of replicas in its spec.replicas, and the controller makes sure that exact number of pod instances are running by creating or terminating pods as needed. You can adjust this count to scale the app either by changing the replicas value (for example, via a CLI command or by editing the deployment) or by using autoscaling, but the Deployment is the resource that enforces the replica count. By contrast, a Service handles networking to pods, a Route exposes the service outside the cluster, and a ConfigMap stores configuration data—none of these directly set the number of pod replicas.

10. What is the command to create a configmap cm3 with the literal value DB_HOST=localhost?

- A. oc create cm3 --from-literal DB_HOST=localhost
- B. oc create configmap cm3 --from-file /tmp/values.txt
- C. oc create configmap cm3 --from-literal DB_HOST=localhost**
- D. oc create configmap cm3 --from-file /etc/db_host.txt

ConfigMaps can be created from literal key-value pairs using the --from-literal flag. To create a ConfigMap named cm3 that contains DB_HOST with value localhost, use the resource type and supply the literal: oc create configmap cm3 --from-literal DB_HOST=localhost. Using --from-file would pull data from a file (creating keys from filenames or file contents), which isn't the explicit DB_HOST=localhost value you need, and omitting the resource type would make the command invalid because OpenShift expects the resource to know what to create.

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://redhatopenshiftdevex288.examzify.com>

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

SAMPLE