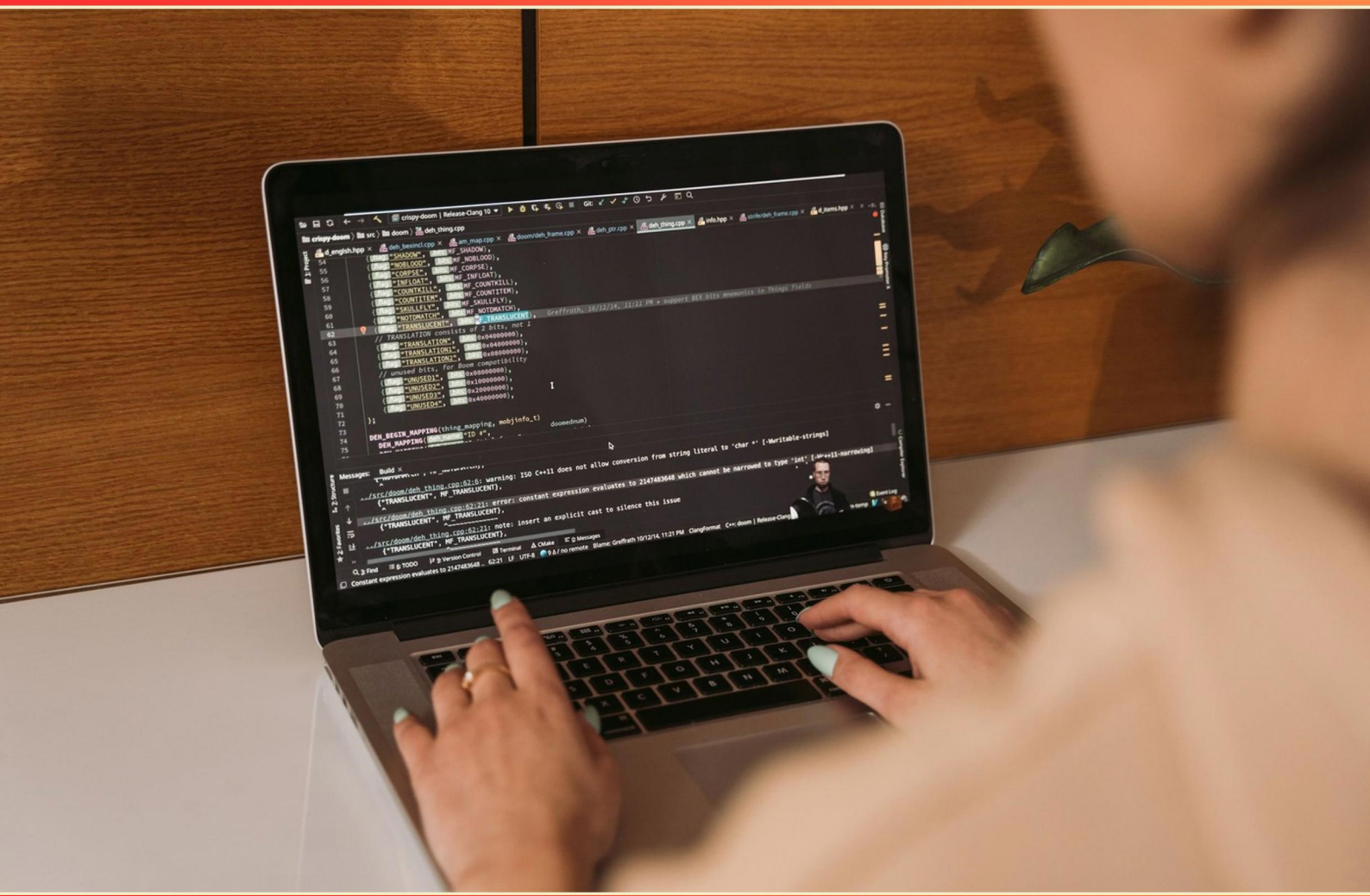
RED HAT OPENSHIFT DEVELOPER EX288 PRACTICE EXAM (SAMPLE)

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 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. Where do readiness probes go in a deployment spec?

A. spec.containers.readinessprobe

B. spec.template.containers.readinessProbe

C. spec.readinessProbe

D. status.readiness

2. Which command configures the build config phpbuild to trigger whenever the base image newapp/php:1 changes?

A. oc set trigger bc/phpbuild --from-image=newapp/php:1

B. oc set trigger bc/phpbuild --from-image=newapp/php:2

C. oc set triggers bc/phpbuild --from-image=newapp/php:1

D. oc set trigger bc/phpbuild --from-image=newapp/php:0

3. How would you verify that a newly created Route is correctly exposing your application?

A. Open the Route in a browser using the cluster IP.

B. Retrieve the Route host with oc get routes, wait for DNS to resolve, then curl the Route host and check the response.

C. Use oc describe route and assume it's exposed.

D. Use oc expose service to create a new route.

4. Which command creates an app named myjava from template dev/lampstack in namespace dev with PORT=8080?

A. oc new-app --template dev/lampstack -p PORT=8080 --name myjava --namespace dev

B. oc new-app --template dev/lampstack -p PORT=8080 --name myjava -n dev

C. oc new-app --template dev/lampstack -p PORT=8080 --name myjava

D. oc new-app --template dev/lampstack -p PORT=8080 --name myjava -n dev

5. After processing a Template with oc process, which command can you use to create the resources described by the output?

A. oc apply -f -

B. oc create -f-

C. oc apply -f template.yaml

D. oc login

SAMPLE

6. What is the purpose of a Security Context Constraint (SCC) in OpenShift?

A. To configure internal network policies.

B. To control what a pod is allowed to do, such as user IDs, privileged mode, and volume types.

C. To store secrets.

D. To encrypt data at rest.

7. What is the purpose of SecurityContextConstraints in OpenShift?

A. SCC defines pod security policy (user, group, privileges, allowed host paths, SELinux, etc.). OpenShift uses SCC to enforce security restrictions on pods.

B. SCC defines network policies for pods.

C. SCC defines resource quotas for namespaces.

D. SCC defines storage classes.

8. What is a PodDisruptionBudget (PDB), and when would you create one?

A. A PodDisruptionBudget defines the minimum number of available pods during voluntary disruptions, used during node drains or upgrades to maintain availability.

B. A PDB sets network policies for pods during disruptions.

C. A PDB configures storage policies.

D. A PDB defines the image pull policy.

9. How do labels and selectors connect between Services, Pods, and Deployments in OpenShift?

A. Labels are key-value pairs attached to objects; Deployments select pods via label selectors; Services route to pods by label selectors.

B. Labels are metadata used only for humans; Deployments ignore selectors.

C. Pods use selectors to find Services.

D. Services define labels; Deployments ignore them.

10. How do you run a locally built s2i image named php:1 for testing?

A. podman run -d -u 1234 php:1

B. docker run -d -u 1234 php:1

C. podman run -d -u1234 php:1

D. kubectl run php --image=php:1

SAMPLE

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. Where do readiness probes go in a deployment spec?

A. spec.containers.readinessprobe

B. spec.template.containers.readinessProbe

C. spec.readinessProbe

D. status.readiness

Readiness probes are defined for each container inside the pod that a Deployment creates. You specify them in the container's configuration within the pod template, not at the top level of the Deployment or in the status. In practice, you place the readinessProbe under the container in the pod spec of the template, i.e., spec.template.spec.containers[].readinessProbe. This means every container can have its own probe to report when it's ready to accept traffic. Example snippet: spec: template: spec: containers: - name: app image: myimage readinessProbe: httpGet: path: /health port: 80 initialDelaySeconds: 5 periodSeconds: 10 So the key idea is: readiness probes live inside each container in the pod template, not at the Deployment or status level.

2. Which command configures the build config phpbuild to trigger whenever the base image newapp/php:1 changes?

A. oc set trigger bc/phpbuild --from-image=newapp/php:1

B. oc set trigger bc/phpbuild --from-image=newapp/php:2

C. oc set triggers bc/phpbuild --from-image=newapp/php:1

D. oc set trigger bc/phpbuild --from-image=newapp/php:0

OpenShift builds can automatically start when the image they rely on changes. This is done by configuring an ImageChange trigger on the BuildConfig, specifying which image stream and tag to watch. The correct command targets the BuildConfig by name and attaches a trigger that watches the image newapp/php:1. When that image tag updates (a new image is pushed for that tag), a new build for phpbuild will be triggered automatically. The other options would either watch a different tag (which wouldn't respond to changes to tag 1) or use a variant of the command that doesn't apply the same trigger in this context.

3. How would you verify that a newly created Route is correctly exposing your application?

A. Open the Route in a browser using the cluster IP.

B. Retrieve the Route host with oc get routes, wait for DNS to resolve, then curl the Route host and check the response.

C. Use oc describe route and assume it's exposed.

D. Use oc expose service to create a new route.

End-to-end verification that a Route exposes the application by resolving the Route's host and making a real HTTP request to that hostname. The Route object defines an externally reachable host through the cluster's router, so the true test is to confirm you can reach that host from outside or from a client machine and receive the expected response. First, list the routes to obtain the exact host name assigned to your Route, which you can then use to perform a request. If you need to, wait for the DNS for that host to propagate so the name actually resolves to the router's IP. Once DNS resolves, curl the Route host (using the appropriate protocol, http or https) and verify you see the expected status code and content. This end-to-end check confirms the Route is correctly exposing the application because it validates the actual network path from a client to the service through the OpenShift router. Opening the Route via the cluster IP isn't reliable for external access since Routes are designed to be reached by DNS-hostnames managed by the cluster's DNS, not by internal cluster IPs. Simply describing the Route doesn't prove reachability, you must verify the real connection works. Creating a Route with the expose command is how you set up exposure, not how you verify it.

SAMPLE

4. Which command creates an app named myjava from template dev/lampstack in namespace dev with PORT=8080?

A. oc new-app --template dev/lampstack -p PORT=8080 --name myjava --namespace dev

B. oc new-app --template dev/lampstack -p PORT=8080 --name myjava -n dev

C. oc new-app --template dev/lampstack -p PORT=8080 --name myjava

D. oc new-app --template dev/lampstack -p PORT=8080 --name myjava -n dev

The command is using a template to create an app, while explicitly setting the port, the application name, and the target namespace. It pulls the template dev/lampstack, passes the parameter PORT=8080 so the app will use that port, names the resulting application myjava, and places the resources into the dev namespace with the short form -n for namespace. This combination directly satisfies "create an app named myjava from template dev/lampstack in namespace dev with PORT=8080." The long form for the namespace (--namespace dev) is also valid in other contexts, but the version using -n dev is a common, concise way to specify the target project in OpenShift.

5. After processing a Template with oc process, which command can you use to create the resources described by the output?

A. oc apply -f -

B. oc create -f-

C. oc apply -f template.yaml

D. oc login

Processing a Template yields a set of resource definitions in YAML/JSON, which oc process writes to stdout. To actually create those resources, feed that output into a create operation that reads manifests from standard input. oc create -f - reads the processed output from stdin and creates all the resources in the cluster in one step. The other options don't fit this flow: login is just authentication, and while applying the processed output with oc apply -f - could also create resources, the chosen command directly mirrors the common template-to-resource creation pattern.

6. What is the purpose of a Security Context Constraint (SCC) in OpenShift?

A. To configure internal network policies.

B. To control what a pod is allowed to do, such as user IDs, privileged mode, and volume types.

C. To store secrets.

D. To encrypt data at rest.

Security Context Constraints define the boundary of what a pod is allowed to do at runtime. They are applied by the OpenShift admission controller to enforce security settings for pods, ensuring workloads run with constrained permissions unless explicitly permitted. A SCC controls aspects like which user ID a container can run as, whether privileged mode is allowed, if privilege escalation is permitted, and which volume types or mount options can be used. This helps prevent actions that could compromise nodes or other tenants, such as running as root or mounting sensitive host volumes, by requiring pods to operate within approved security settings. Network policies govern how pods can communicate with each other, not the security context of a pod. Secrets storage is about securely managing sensitive data, not enforcing runtime security constraints. Encryption at rest is a cluster-wide data protection feature, separate from pod-level security constraints.

SAMPLE

7. What is the purpose of SecurityContextConstraints in OpenShift?

A. SCC defines pod security policy (user, group, privileges, allowed host paths, SELinux, etc.). OpenShift uses SCC to enforce security restrictions on pods.

B. SCC defines network policies for pods.

C. SCC defines resource quotas for namespaces.

D. SCC defines storage classes.

SecurityContextConstraints define how a pod can run in OpenShift by specifying security-related settings for pods and containers. They control aspects like which users and groups can run, whether privileged mode is allowed, SELinux context, capabilities, runAsUser/runAsNonRoot strategies, and how the filesystem and host resources can be accessed (for example, allowed host paths and readOnlyRootFilesystem). When a user creates a pod, OpenShift checks the pod's securityContext against the SCCs bound to that user or service account; only if the pod complies with an allowed SCC is it permitted to run. This is why this option best describes the purpose: OpenShift uses SecurityContextConstraints to enforce security restrictions on pods. Network policies govern pod-to-pod traffic, not securityContextConstraints; resource quotas manage resource usage in namespaces; storage classes define dynamic provisioning of storage.

8. What is a PodDisruptionBudget (PDB), and when would you create one?

A. A PodDisruptionBudget defines the minimum number of available pods during voluntary disruptions, used during node drains or upgrades to maintain availability.

B. A PDB sets network policies for pods during disruptions.

C. A PDB configures storage policies.

D. A PDB defines the image pull policy.

A PodDisruptionBudget is a Kubernetes object that defines how many pods can be disrupted during voluntary events, so your application stays available during maintenance. It targets a set of pods (usually by labels) and specifies a threshold like minimum available or maximum unavailable. During actions such as node drains or cluster upgrades, the system will only evict pods if doing so would not drop below that minimum. For example, with five replicas and a minimum available of four, at most one pod can be evicted during a drain. This is precisely what keeps your service online while you perform maintenance. It isn't about network policies, storage policies, or image pull behavior, which is why the other options don't fit.

SAMPLE

9. How do labels and selectors connect between Services, Pods, and Deployments in OpenShift?

A. Labels are key-value pairs attached to objects; Deployments select pods via label selectors; Services route to pods by label selectors.

B. Labels are metadata used only for humans; Deployments ignore selectors.

C. Pods use selectors to find Services.

D. Services define labels; Deployments ignore them.

Labels are key-value pairs attached to objects to identify and group them. A Deployment defines how many Pods to run and uses a label selector to decide which Pods belong to that Deployment; the Pod template it creates must carry labels that match this selector so the Deployment can manage those Pods. A Service, on the other hand, doesn't route by itself to any Pod; it uses its own label selector to choose which Pods should receive traffic. When the Pods have the same labels that the Deployment and the Service selectors use, the Deployment reliably manages the intended Pods and the Service can route requests to those Pods. In OpenShift, this pattern is how you connect workloads to each other: the labels form a contract that ties the Deployment's managed Pods to the Service that exposes them. If the labels don't align, the Service will have no endpoints, or the Deployment will not manage the correct Pods.

10. How do you run a locally built s2i image named php:1 for testing?

A. podman run -d -u 1234 php:1

B. docker run -d -u 1234 php:1

C. podman run -d -u1234 php:1

D. kubectl run php --image=php:1

When testing a locally built s2i image, you want to run a container from that image in the background and, if the image is designed to run as a non-root user, specify that user. Podman fits this scenario well for local, rootless testing. The image tag php:1 refers to your locally built image. The -d flag starts the container in detached mode, so it runs in the background. The -u1234 flag sets the user to UID 1234, aligning the container's execution with the non-root user the image expects. Using the compact form -u1234 is just another valid way to pass the user value to Podman. This setup lets you quickly verify the image's behavior in a realistic, non-root environment, which is why this option is the best fit for local testing.

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://redhatopenshiftdevex288.examzify.com

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

SAMPLE