DOCKER FOUNDATIONS PROFESSIONAL CERTIFICATE PRACTICE TEST (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR THE FULL VERSION STUDY GUIDE

https://dockerfoundationspro.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 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.

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. What function does the `VOLUME` instruction serve in a Dockerfile?

A. Designates a path for data backup

B. Establishes a connection to external databases

C. Creates a mount point and marks it for external data

D. Sets disk usage limits for the container

2. What is accomplished by the command 'docker-compose up'?

A. Stops running Docker services

B. Starts the services defined in a Docker Compose file

C. Deploys a Docker Swarm service

D. Creates a new Docker image

3. What command is used to create a new Docker container?

A. docker build

B. docker create

C. docker run

D. docker start

4. Which command is used to copy files from the host machine to a running container?

A. docker send

B. docker cp

C. docker pull

D. docker transfer

5. How can you see the logs of a specific Docker container?

A. docker log [container_id]

B. docker view [container_id]

C. docker inspect [container_id]

D. docker logs [container_id]

6. Which command is used to start a new Docker container in the foreground?

A. docker run -d [image_name]

B. docker start -f [container_name]

C. docker run [image_name]

D. docker create [image_name]

SAMPLE

7. What is one of the primary purposes of Docker Swarm?

A. To simplify the building of Docker images

B. To manage the deployment of multi-container applications

C. To track container usage and limits

D. To facilitate Docker CLI commands

8. What is the function of the ENTRYPOINT instruction in a Dockerfile?

A. To set environment variables

B. To specify the command that runs when a container starts

C. To expose ports for the container

D. To create a Docker image from the Dockerfile

9. What is meant by "container image"?

A. A compressed archive of Docker containers

B. A snapshot of a filesystem and parameters

C. A running instance of a Docker application

D. A log of Docker container events

10. How can you specify environment variables when running a container?

A. Using the '--env' flag in the 'docker create' command

B. Using the '-e' flag in the 'docker run' command

C. By editing the Dockerfile directly

D. By modifying the .dockerignore file

SAMPLE

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. What function does the `VOLUME` instruction serve in a Dockerfile?

A. Designates a path for data backup

B. Establishes a connection to external databases

C. Creates a mount point and marks it for external data

D. Sets disk usage limits for the container

The correct choice highlights the key purpose of the `VOLUME` instruction in a Dockerfile, which is to create a mount point within the container's filesystem that is designated for external data storage. When this instruction is used, it enables the container to persist data outside of its own filesystem, allowing it to maintain state even if the container itself is stopped or removed. This functionality is essential for applications that require data to be saved between container sessions, such as databases or any application that needs to retain user-generated data. By marking a path with the `VOLUME` instruction, Docker creates a mount point that can be linked to storage options like host directories, cloud storage, or data volumes. This capability supports data portability and facilitates recovery from container restarts. The other options do not accurately describe the role of the `VOLUME` instruction. Designating a path for data backup may be a goal when using volumes, but it is not the primary function of the instruction. Establishing connections to external databases is related to network configurations and not the specific purpose of `VOLUME`. Setting disk usage limits is not something controlled by the `VOLUME` instruction; instead, it could involve different Docker resource management features that are unrelated to how volumes are mounted.

2. What is accomplished by the command 'docker-compose up'?

A. Stops running Docker services

B. Starts the services defined in a Docker Compose file

C. Deploys a Docker Swarm service

D. Creates a new Docker image

The command 'docker-compose up' is essential for managing multi-container Docker applications. When executed, it initiates the services defined in a Docker Compose file (usually named `docker-compose.yml`). This file outlines the configuration for each service, including details like the image to use, ports to expose, environment variables, networks, and volumes. By running 'docker-compose up', Docker Compose reads this configuration and starts all the specified services in the appropriate order, ensuring that any service dependencies are respected. If the images for those services don't already exist, Docker Compose will automatically pull them from a registry or build them if specified in the compose file. This command effectively sets up the entire environment needed for the application to run, making it straightforward to manage complex, multi-container setups with a single command.

SAMPLE

3. What command is used to create a new Docker container?

A. docker build

B. docker create

C. docker run

D. docker start

The command used to create a new Docker container is "docker run." This command not only creates the container but also starts it immediately. When you execute "docker run," it takes an image as an argument and creates a new container from that image, effectively combining the actions of both creating and starting a container into a single command. Using "docker run" is very common in workflows, as it simplifies the process of instantiating containers from images. Additionally, this command allows you to specify various options and configurations for the new container, including environment variables, port mappings, and volume mounts at the time of creation. In contrast, the command "docker create" only creates a new container without starting it, while "docker start" is used to start an existing stopped container. The command "docker build" is primarily used to build a Docker image from a Dockerfile and does not pertain directly to container creation.

4. Which command is used to copy files from the host machine to a running container?

A. docker send

B. docker cp

C. docker pull

D. docker transfer

The command used to copy files from the host machine to a running container is "docker cp." This command allows users to specify the source path on the host and the destination path inside the container, enabling seamless transfer of files between the two environments. The utility of this command lies in its ability to facilitate file management within containers, which may not have direct access to the host's filesystem, thus enhancing the flexibility of container operations. In contrast to other commands, "docker cp" directly deals with file transfers, making it a straightforward and effective solution for moving files. The other options do not exist in Docker's command set for this purpose. For instance, there is no "docker send" or "docker transfer" command associated with file transfer functionalities, and "docker pull" is specifically designed for downloading Docker images from a registry, not copying files to a running container.

5. How can you see the logs of a specific Docker container?

A. docker log [container_id]

B. docker view [container_id]

C. docker inspect [container_id]

D. docker logs [container_id]

To view the logs of a specific Docker container, the correct command is formed as "docker logs [container_id]". This command is specifically designed to fetch and display the logs that are generated by the container specified by the container ID or name. When a Docker container runs, it outputs logs that provide crucial information about the application's behavior, errors, and other runtime details. Using docker logs allows users to access this information, which is important for troubleshooting and monitoring the application. The other options do not fulfill this specific function. For instance, "docker log" is not a valid command, while "docker view" is not a recognized Docker command at all. "Docker inspect" is useful for retrieving detailed information about a container's configuration and state but does not provide log output. Thus, using "docker logs" is the standard and correct method to view a container's logs.

SAMPLE

6. Which command is used to start a new Docker container in the foreground?

A. docker run -d [image_name]

B. docker start -f [container_name]

C. docker run [image_name]

D. docker create [image_name]

The command that starts a new Docker container in the foreground is indeed "docker run [image_name]." This command is essential when you want to create and start a new container directly from a specified image, allowing you to interact with the container's standard input and output through your terminal. When you use this command without the "-d" option, it operates in the foreground, meaning you will see the logs and output directly in your terminal window. The other commands do not fulfill the requirement of starting a new container in the foreground. "docker run -d [image_name]" starts a container in detached mode, which runs in the background. "docker start -f [container_name]" is used to start an already existing stopped container and does not create a new one. Finally, "docker create [image_name]" only creates a new container from the specified image but does not start it, so you would need a subsequent command like "docker start [container_name]" to actually run the container. Therefore, the correct choice is to use "docker run [image_name]" to initiate a new Docker container and interact with it directly in the terminal.

7. What is one of the primary purposes of Docker Swarm?

A. To simplify the building of Docker images

B. To manage the deployment of multi-container applications

C. To track container usage and limits

D. To facilitate Docker CLI commands

One of the primary purposes of Docker Swarm is to manage the deployment of multi-container applications. Docker Swarm provides a native clustering and orchestration solution for Docker containers, enabling users to easily deploy and manage applications that consist of multiple interdependent containers across a distributed cluster of machines. This capability is crucial in modern software development and deployment because applications often involve multiple services that need to communicate with each other. Docker Swarm handles service discovery, load balancing, scaling, and failover for these multi-container applications, allowing developers and system administrators to focus on building and deploying their applications without having to manage the underlying complexities of container orchestration manually. The other options focus on different aspects of Docker's functionality. Simplifying the building of Docker images is managed by tools like Dockerfile and Docker build, which are separate from the orchestration features of Swarm. Tracking container usage and limits pertains to resource monitoring and management, which is handled through different Docker commands and configurations, rather than Swarm itself. Facilitating Docker CLI commands refers to the ease of using the command-line interface to issue instructions to Docker but does not address the specific role of Swarm in managing containerized applications.

SAMPLE

8. What is the function of the ENTRYPOINT instruction in a Dockerfile?

A. To set environment variables

B. To specify the command that runs when a container starts

C. To expose ports for the container

D. To create a Docker image from the Dockerfile

The ENTRYPOINT instruction in a Dockerfile is crucial for defining what command will be executed when a container is started from the image that the Dockerfile creates. This instruction establishes the main process that runs in the container, ensuring that the specified application or script is executed upon container launch. By using ENTRYPOINT, users can setup the container to behave like a standalone executable, making it intuitive for others to run the container with default behavior. While there are different ways to specify commands in Dockerfiles, using ENTRYPOINT provides flexibility and control over how the command is executed. It can also be used in conjunction with the CMD instruction, which can supply default arguments to the ENTRYPOINT command. This means that the ENTRYPOINT can be fixed, while CMD allows some customization by users when they run the container. In contrast, setting environment variables, exposing ports, or creating a Docker image from the Dockerfile are separate functionalities that serve distinct purposes in containerization and image management. Understanding the role of each instruction helps in effectively building and managing Docker containers.

9. What is meant by "container image"?

A. A compressed archive of Docker containers

B. A snapshot of a filesystem and parameters

C. A running instance of a Docker application

D. A log of Docker container events

A "container image" refers to a snapshot of a filesystem and parameters that define the software environment for a Docker container. This image serves as a blueprint for creating Docker containers, encapsulating everything needed to run an application, including libraries, binaries, and configuration files. When a container is instantiated from this image, it operates in a lightweight, isolated environment, which ensures that the application runs consistently across different environments. The concept of a container image allows developers to package applications and their dependencies into a self-sufficient unit, making deployment and scaling significantly easier. By using a layered filesystem, images allow for efficient storage and sharing, where only changes between layers need to be stored if a new image is created based on an existing one. The other options do not accurately describe what a container image is. A compressed archive of Docker containers refers to a different format and stage in Docker workflow. A running instance of a Docker application describes a container, not the image itself. A log of Docker container events pertains to monitoring and tracking activity rather than defining the structure or purpose of an image. Hence, the definition as a snapshot of a filesystem and parameters is the most accurate description of a container image.

SAMPLE

10. How can you specify environment variables when running a container?

A. Using the '--env' flag in the 'docker create' command

B. Using the '-e' flag in the 'docker run' command

C. By editing the Dockerfile directly

D. By modifying the .dockerignore file

Specifying environment variables when running a container is commonly done using the '-e' flag in the 'docker run' command. This approach allows you to define environment variables directly in the command line when initiating a container, providing flexibility for dynamically setting configurations that the application inside the container might need at runtime. For instance, the command `docker run -e ENV_VAR_NAME=value image_name` sets the environment variable `ENV_VAR_NAME` to `value` within the container. This is particularly useful for sensitive data like API keys or configuration settings that should not be hard-coded into the image or source code. While there are other methods to set environment variables, such as using the '--env' flag in the 'docker create' command, these are less common for standard runtime practices. Editing the Dockerfile or modifying the .dockerignore file does not facilitate the direct runtime specification of environment variables; they serve completely different purposes in the Docker workflow. The Dockerfile determines the configuration of the Docker image itself, while the .dockerignore file excludes files from being sent to the Docker daemon during the image build process. Therefore, using the '-e' flag in the 'docker run' command is the standard, most straightforward method for setting environment variables while running a container

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://dockerfoundationspro.examzify.com

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

SAMPLE