

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.

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. Which command is used to display the version of Docker?**
 - A. docker info
 - B. docker version
 - C. docker status
 - D. docker show version
- 2. How does a Docker image differ from a Docker container?**
 - A. An image is an executable package, while a container is a template
 - B. An image is a read-only template used to create containers
 - C. A container contains multiple images, while an image is always singular
 - D. An image cannot be modified, while a container can
- 3. What is the purpose of a Dockerfile?**
 - A. To manage Docker containers
 - B. To define services and networks in Docker Compose
 - C. To automate the creation of a Docker image
 - D. To monitor resource usage of containers
- 4. What does the Docker command 'docker ps' do?**
 - A. It creates a new container
 - B. It lists the running containers
 - C. It removes a container
 - D. It builds an image
- 5. What information does the docker info command display?**
 - A. Information about the running containers
 - B. System-wide information about the Docker installation
 - C. Status of the Docker services
 - D. Details of the network configuration
- 6. Which command is used to view all available Docker commands?**
 - A. docker commands
 - B. docker help
 - C. docker list commands
 - D. docker command list

7. If a container has exited, which command can you use to see its ID?

- A. `docker ps`
- B. `docker inspect`
- C. `docker logs`
- D. `docker ps -a`

8. What is an environment variable in a Docker container?

- A. A secret key stored securely
- B. A variable used for container networking
- C. A setting that configures the behavior of the container
- D. A visual representation of data in a Docker container

9. How do you update a running Docker service in Swarm mode?

- A. Use the '`docker service upgrade`' command
- B. Use the '`docker service change`' command
- C. Use the '`docker service update`' command
- D. Use the '`docker service refresh`' command

10. What command is used to stop a running Docker container?

- A. `docker stop [container_id]`
- B. `docker terminate [container_id]`
- C. `docker kill [container_id]`
- D. `docker end [container_id]`

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. Which command is used to display the version of Docker?

- A. docker info
- B. docker version**
- C. docker status
- D. docker show version

The command used to display the version of Docker is "docker version." This command provides detailed information about the Docker installation, including the version numbers of the client and server as well as the Git commit ID. It is a straightforward and effective way to verify which version of Docker is currently running on your system, which is critical for ensuring compatibility with images and containers, as well as for troubleshooting any issues that may arise. In contrast, while "docker info" provides a wealth of information about the Docker environment, including system-wide information and configuration details, it does not specifically focus just on the version. Other options like "docker status" and "docker show version" do not exist as recognized commands in Docker CLI, making them unsuitable for this task. Hence, using "docker version" is the most reliable method for quickly determining the version of your Docker installation.

2. How does a Docker image differ from a Docker container?

- A. An image is an executable package, while a container is a template
- B. An image is a read-only template used to create containers**
- C. A container contains multiple images, while an image is always singular
- D. An image cannot be modified, while a container can

A Docker image fundamentally serves as a blueprint or a read-only template for creating instances known as containers. Images encapsulate everything necessary for running software, such as the code, runtime, libraries, environment variables, and configurations. When a Docker image is executed, it generates a container that runs an isolated instance of the software defined within that image. This distinction is crucial in understanding containerization because it highlights that images themselves do not perform tasks; rather, they are utilized to spawn containers, which are the operational entities that actually run processes. The correct choice underscores the read-only nature of images and their function in creating containers. This is a core aspect of how Docker operates, enabling efficient deployment and management of applications through isolation and abstraction. In contrast to this, there are misunderstandings in the other options. The idea of an image being a template can mislead someone into thinking it's similar to a container, which is an active runtime instance rather than just a static source. Additionally, the incorrect notion that a container can encompass multiple images misrepresents the architecture of Docker, which distinguishes clearly between individual images and instant containers created from them. Lastly, while it's true that images cannot be modified after their creation, the essence of the relationship between an image and a

3. What is the purpose of a Dockerfile?

- A. To manage Docker containers
- B. To define services and networks in Docker Compose
- C. To automate the creation of a Docker image**
- D. To monitor resource usage of containers

The purpose of a Dockerfile is to automate the creation of a Docker image. A Dockerfile is essentially a set of instructions that Docker uses to build an image. It contains the steps required to assemble the image, including specifying a base image, running commands, copying files, and setting environment variables. This automation means that developers can define the environment their applications will run in with repeatability and without manual intervention. Using a Dockerfile streamlines the process of deploying applications, fosters consistency across different environments, and simplifies version control of the application environment. When the build process is initiated, Docker reads the Dockerfile line by line to create the final image, which can then be used to run containers. The other options refer to different aspects of Docker management, but they do not capture the specific function of a Dockerfile. For example, managing Docker containers pertains to runtime operations, while defining services and networks relates to Docker Compose, which is a tool designed for multi-container Docker applications. Monitoring resource usage of containers involves other tools and does not relate to the functionality of a Dockerfile.

4. What does the Docker command 'docker ps' do?

- A. It creates a new container
- B. It lists the running containers**
- C. It removes a container
- D. It builds an image

The command 'docker ps' is specifically designed to list all the currently running Docker containers on a host. When executed, it provides a snapshot of active containers, displaying important information such as container IDs, names, and the status of the containers, including how long they have been running. This command is essential for users to monitor and manage their running applications effectively since it gives quick visibility into what is active at any given moment. The context of each of the incorrect options helps clarify the specific function of 'docker ps': - Creating a new container is done using the 'docker run' command, which both creates and starts a container from a specified image. - Removing a container would utilize the 'docker rm' command, which is not related to listing any current container's status. - Building an image is accomplished using the 'docker build' command, which compiles source files into a Docker image, unrelated to the state of current containers. Understanding the correct command usage is vital for efficient Docker management, reinforcing the importance of 'docker ps' in routine operations.

5. What information does the docker info command display?

- A. Information about the running containers
- B. System-wide information about the Docker installation**
- C. Status of the Docker services
- D. Details of the network configuration

The `docker info` command provides a comprehensive overview of the Docker installation on your system. It presents system-wide information, including details about the Docker daemon, the number of containers (both running and stopped), images, storage driver, and more. This command serves as a useful diagnostic tool, allowing users to quickly gather essential information about their Docker environment, such as the version of Docker, the number of volumes, and the orchestration system in use, if any. This broad scope of information is what sets the correct option apart from the others. While some options may reference specific components of Docker, such as running containers or network configuration, the `docker info` command specifically encompasses an overall snapshot of the Docker installation rather than focusing on specific operational elements. Therefore, its utility in providing a holistic view of the Docker setup makes this the correct answer.

6. Which command is used to view all available Docker commands?

- A. `docker commands`
- B. `docker help`**
- C. `docker list commands`
- D. `docker command list`

Using the command `'docker help'` provides a comprehensive view of all available Docker commands. When executed, this command displays a list of commands along with a brief description of what each command does. It serves as a quick reference guide for users to familiarize themselves with Docker's extensive command set, including both high-level commands and those specific to particular operations. In contrast, the other options do not represent valid commands within Docker. There is no `'docker commands'`, `'docker list commands'`, or `'docker command list'` command in Docker's CLI. Hence, they would not provide any output related to available commands, reinforcing the correctness of `'docker help'` as the right choice for viewing all the commands within Docker.

7. If a container has exited, which command can you use to see its ID?

- A. `docker ps`
- B. `docker inspect`
- C. `docker logs`
- D. `docker ps -a`**

When a container has exited, you can use the command that lists all containers, regardless of their status, to see its ID. This command provides a complete overview, including running, stopped, and exited containers. By using this command, you can effectively manage containers, check their statuses, and retrieve the specific IDs of those that have exited. This is particularly useful for debugging or when you need to restart or remove the container. In contrast, the command that only lists running containers will not show exited ones, while the commands for inspecting a container or viewing logs are used for different purposes. Inspecting is for getting detailed information about a current container, and checking logs is to view the output of a container's process, neither of which relate to simply obtaining the ID of an exited container directly.

8. What is an environment variable in a Docker container?

- A. A secret key stored securely
- B. A variable used for container networking
- C. A setting that configures the behavior of the container**
- D. A visual representation of data in a Docker container

An environment variable in a Docker container serves as a setting that configures the behavior of the container. These variables allow you to customize the configuration of applications running inside the container without changing the application code. For instance, you can set environment variables to specify database connection strings, API keys, or application modes (like development or production). By defining these variables when you create and run the container, you enable the same image to run in different environments with different configurations. This flexibility is particularly valuable in scenarios where you want to deploy the same application across various environments without modifying the underlying image. The application inside the container can access these environment variables, adapting its runtime behavior based on the values provided. Other listed options describe different concepts that are not applicable to environment variables specifically. For example, while secret keys are important for security, they refer to a specific use case rather than the general function of an environment variable. Similarly, container networking is a broader aspect of Docker's functionality, not exclusively defined by environment variables. Lastly, visual representation of data does not relate to the concept of environment variables, as they are not visual elements but rather key-value pairs stored in the container's environment.

9. How do you update a running Docker service in Swarm mode?

- A. Use the 'docker service upgrade' command
- B. Use the 'docker service change' command
- C. Use the 'docker service update' command**
- D. Use the 'docker service refresh' command

To update a running Docker service in Swarm mode, the appropriate command is 'docker service update.' This command allows you to modify various parameters of the service, such as the image version, environment variables, resource limits, and more, while the service is actively running. It ensures that the changes are applied seamlessly across the service's tasks, handling the update process with minimal downtime. This command supports a variety of options that can specify how the update should be conducted. For instance, you can control the rollback behavior if needed or define how many tasks are updated at once, which is particularly useful for managing availability during the transition. The other options provided do not correspond to valid Docker commands for updating services in Swarm mode. For example, 'docker service upgrade' is not a recognized command, as the correct term for updating services is 'update.' Similarly, 'docker service change' and 'docker service refresh' are not part of the Docker CLI and thus cannot be used to perform service updates in any context.

10. What command is used to stop a running Docker container?

- A. docker stop [container_id]
- B. docker terminate [container_id]
- C. docker kill [container_id]
- D. docker end [container_id]

The command to stop a running Docker container is "docker stop [container_id]". This command gracefully stops the container, allowing it to complete its current processes before shutting down. By default, it sends a SIGTERM signal to the main process in the container, which can then perform cleanup operations if necessary. If the process does not terminate within a specified timeout period, a SIGKILL signal is sent to forcibly stop the container. This approach is preferred for stopping containers because it ensures that any ongoing operations are completed properly, which can help prevent data loss or corruption. The other choices do not represent the correct way to stop a container: "docker terminate" is not a valid Docker command, "docker kill" forcibly stops a container by sending the SIGKILL signal immediately, which does not allow for graceful shutdown, and "docker end" is also not a recognized command in Docker's CLI. Thus, "docker stop" is the appropriate command for stopping a running container in a safe and orderly manner.

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