DOCKER CERTIFIED ASSOCIATE (DCA) CERTIFICATION PRACTICE TEST (SAMPLE)

STUDY GUIDE

SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR THE FULL VERSION STUDY GUIDE

https://dockerassociate.examzify.com

Table of Contents

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

1. What command is used to rotate a Docker swarm unlock-key?

A. docker swarm unlock-key --update

B. docker swarm unlock-key --rotate

C. docker swarm key --rotate

D. docker swarm rotate-key

2. What Linux feature allows Docker containers to listen on ports lower than 1024 without root privileges?

A. Namespaces

B. Capabilities

C. Control Groups

D. Privileges

3. Which flag is used with docker inspect to return specific fields?

A. --filter

B. --select

C. --format

D. --fields

4. What command generates a new Dockerfile?

A. docker create

B. docker init

C. dockerfile new

D. There is no specific command; it must be created manually

5. How can Tracy prevent her development team from overwriting images in a Docker Trusted Registry?

A. By using a dedicated namespace for each image

B. By marking the repository as immutable

C. By enforcing a tagging convention

D. By resetting the repository password regularly

6. What does the 'docker ps' command display?

A. Information about Docker networks

B. Information about running containers

C. Information about Docker images

D. Information about Docker volumes

7. What is used to specify service dependencies in Docker Compose?

A. JSON file

B. YAML file

C. XML file

D. INI file

8. How do you check the health status of a running container?

A. docker status [container_id]

B. docker health [container_id]

C. docker inspect [container_id]

D. docker ps -q --filter "health=healthy"

9. Which command is used to remove all stopped containers?

A. docker rm --all

B. docker container prune

C. docker stop all

D. docker delete --stopped

10. What is the command used for managing Docker networks?

A. docker network

B. docker connect

C. docker bridge

D. docker route

Answers

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

Explanations

1. What command is used to rotate a Docker swarm unlock-key?

A. docker swarm unlock-key --update

B. docker swarm unlock-key --rotate

C. docker swarm key --rotate

D. docker swarm rotate-key

The command to rotate a Docker swarm unlock-key is indeed "docker swarm unlock-key --rotate." This command is specifically designed for the purpose of changing the unlock key associated with a Docker swarm cluster. When you rotate the unlock key, it enhances the security management of the swarm by ensuring that only the most recent key is valid for unlocking the swarm data. This is particularly important in a production environment, where managing access and permissions helps prevent unauthorized access to your swarm and its sensitive data. The presence of the "--rotate" flag indicates that this action is about changing or rotating the key, as opposed to simply retrieving or displaying the current unlock key. This command is essential for maintaining the security posture of Docker swarm clusters, allowing administrators to manage their keys effectively and mitigate potential risks associated with key exposure or compromise.

2. What Linux feature allows Docker containers to listen on ports lower than 1024 without root privileges?

A. Namespaces

B. Capabilities

C. Control Groups

D. Privileges

The correct answer is capabilities. In Linux, capabilities provide a mechanism to separate the privileges of the root user into distinct units, allowing a process to gain specific privileges without being granted full root access. One of these capabilities is the ability to bind to low-numbered ports, which are traditionally restricted to the root user. When Docker containers run as non-root users, they typically cannot listen on ports below 1024 because that is a system-wide security policy. However, by using capabilities, you can enable the container's process to bind to these ports while maintaining a lower security risk compared to running the entire container with full root privileges. This feature enhances security by enabling fine-grained control over what specific actions an application can perform within the container environment, allowing containers to operate more securely and with reduced risk of privilege escalation.

3. Which flag is used with docker inspect to return specific fields?

A. --filter

B. --select

C. --format

D. --fields

The correct flag to use with `docker inspect` for returning specific fields is `--format`. This flag allows users to specify the output format of the information retrieved from the inspection of a Docker object, such as a container or an image. By using Go templating syntax, users can customize the output to display only the information they are interested in, eliminating extraneous details. For example, if you wanted to get the names of all containers, you could use the `--format '{{.Name}}'` option. This level of customization is extremely useful when dealing with large amounts of data or when you're looking to integrate Docker commands into scripts, ensuring that your output is clear and relevant. Other options may sound plausible but do not fulfill the same function. The `--filter` flag, for instance, is used to filter the output based on specific criteria, while `--select` and `--fields` are not valid flags for `docker inspect`. Thus, `--format` stands out as the only option that suitably addresses the need for returning specific fields in a controlled format.

4. What command generates a new Dockerfile?

A. docker create

B. docker init

C. dockerfile new

D. There is no specific command; it must be created manually

The generation of a new Dockerfile does not involve a specific command within Docker itself; instead, it requires manual intervention. A Dockerfile is a text document that contains instructions on how to build a Docker image, including the base image to use, any dependencies to install, and the commands to execute when creating a container from the image. Since Docker does not provide a built-in command such as "docker create," "docker init," or "dockerfile new" for automatically generating a Dockerfile, developers typically create it using a text editor of their choice. This manual process allows developers to customize the Dockerfile according to the specific requirements of their applications or services, ensuring that the image behaves as intended when deployed. The flexibility of manually creating a Dockerfile is significant, as it accommodates a wide variety of use cases and configurations that might not fit into a standardized command structure.

5. How can Tracy prevent her development team from overwriting images in a Docker Trusted Registry?

A. By using a dedicated namespace for each image

B. By marking the repository as immutable

C. By enforcing a tagging convention

D. By resetting the repository password regularly

Marking the repository as immutable is an effective way for Tracy to prevent her development team from overwriting images in a Docker Trusted Registry. An immutable repository ensures that once an image is pushed to the repository, it cannot be altered or deleted. This feature is particularly important in production environments where stability and reliability of container images are critical. By enforcing immutability, Tracy can ensure that all images used in deployment are consistent with the tested versions, thereby minimizing the risk of introducing changes or breaking changes accidentally. Other methods, such as creating a dedicated namespace or enforcing a tagging convention, can help organize images and provide some level of control, but they do not inherently prevent overwriting of existing images. Regularly resetting the repository password may enhance security, but it won't stop team members from overwriting images they have permission to access. Thus, marking the repository as immutable is the most direct and effective solution for Tracy's requirement.

6. What does the 'docker ps' command display?

A. Information about Docker networks

B. Information about running containers

C. Information about Docker images

D. Information about Docker volumes

The 'docker ps' command is specifically designed to show information about currently running containers on a Docker host. When executed, it typically provides details such as the container ID, image used, command being run, creation time, status, ports exposed, and the names of the containers. This command is fundamental for managing containerized applications because it allows users to quickly see which containers are active and their associated metadata. Other choices involve different aspects of Docker's functionality. Information about Docker networks is related to how containers communicate with each other and the outside world, while Docker images refer to the packaged application and its dependencies needed to create a container. Docker volumes pertain to data persistence across container lifecycles and are not displayed by the 'docker ps' command. Thus, the command's focus exclusively on running containers makes it an essential tool for anyone working with Docker in a practical environment.

7. What is used to specify service dependencies in Docker Compose?

A. JSON file

B. YAML file

C. XML file

D. INI file

In Docker Compose, the YAML file format is used to specify service dependencies. YAML (YAML Ain't Markup Language) is preferred for its readability and simplicity, making it easier for developers to define and manage their multi-container applications. In a Docker Compose file, users can define services, networks, and volumes in a structured way, utilizing indentation to represent hierarchical relationships. This structure allows developers to specify how different services depend on each other, enabling features like service orchestration and automatic startup order. For instance, if one service depends on another to be available first, this can be clearly articulated in the YAML configuration. The other file formats mentioned, such as JSON, XML, and INI, are not the standard or conventional choices for Docker Compose. While they may serve different purposes in various applications, Docker Compose specifically utilizes YAML due to its ease of use and the ability to represent the required data hierarchy in a more human-readable format. This choice enhances collaboration among developers when defining complex service dependencies in containerized applications.

8. How do you check the health status of a running container?

A. docker status [container_id]

B. docker health [container_id]

C. docker inspect [container_id]

D. docker ps -q --filter "health=healthy"

To check the health status of a running container, using the command to inspect the container provides a comprehensive overview of its configuration and status. When you execute the `docker inspect [container_id]` command, it returns detailed information in JSON format, including the health status of the container if a health check is defined. This allows the user to see the current state, along with any health check results that have been defined in the container's configuration. In contrast, the other options do not correctly retrieve the health status of a container. The command that attempts to reference a status with `docker status [container_id]` does not exist in Docker's command set. The command `docker health [container_id]` is also not valid, as there is no dedicated health command. Additionally, while `docker ps -q --filter "health=healthy"` filters containers based on their health status for visualization, it does not provide the detailed status for a specific container. Thus, inspecting the container gives the most complete and accurate information regarding its health status.

9. Which command is used to remove all stopped containers?

A. docker rm --all

B. docker container prune

C. docker stop all

D. docker delete --stopped

The command used to remove all stopped containers is indeed "docker container prune." This command efficiently cleans up your Docker environment by removing any stopped containers, freeing up space without affecting any running containers or other elements such as images or volumes. When you run "docker container prune," Docker will prompt you for confirmation and, once confirmed, will delete all containers that are not currently running. This is particularly useful for maintaining a clean environment and ensuring that you do not waste unnecessary resources on containers that are no longer active. In contrast, the other options do not achieve the intended outcome. For example, "docker rm --all" is not a valid command as the correct usage requires specifying individual container IDs or names without a general flag. "docker stop all" does not exist as a command; instead, you would have to specify each container to be stopped. Lastly, "docker delete --stopped" is also incorrect since "delete" is not a recognized Docker command for handling containers. The "docker container prune" command stands out as a powerful tool for Docker management, contributing to a more efficient workspace by removing unnecessary stopped containers in one simple operation.

10. What is the command used for managing Docker networks?

A. docker network

B. docker connect

C. docker bridge

D. docker route

The command used for managing Docker networks is indeed "docker network." This command provides a set of subcommands that allow users to create, inspect, list, and remove networks. This functionality is essential for configuring how containers communicate with each other and the outside world. When utilizing the "docker network" command, you can: - Create new networks for containers to join. - List available networks to understand existing configurations. - Inspect a specific network to gather details about its configuration, including connected containers and network settings. - Remove networks that are no longer needed, which helps in maintaining a clean and efficient Docker environment. This command leverages the Docker networking system, which is fundamental for orchestrating communication between containers and ensuring they can operate effectively in isolation or as part of a larger application. The other options do not pertain specifically to managing Docker networks. "docker connect," for instance, is not a standard command used for network management; it is used for connecting a container to a network. "docker bridge" refers to a specific network driver but is not a command for managing networks. Lastly, "docker route" is not a valid Docker command related to networking. Thus, the "docker network" command stands out as the primary and comprehensive tool for managing Docker

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://dockerassociate.examzify.com

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