

Dockers Menu Practice Test (Sample)

Study Guide



Everything you need from our exam experts!

Copyright © 2025 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 from reliable sources accurate, complete, and timely information about this product.

SAMPLE

Questions

1. What does the ``docker prune`` command do?
 - A. Creates a backup of Docker configurations
 - B. Removes unused data from Docker
 - C. Updates Docker to the latest version
 - D. Starts all stopped containers
2. What is the primary function of the Docker API?
 - A. To create new Docker containers
 - B. To interact programmatically with the Docker daemon
 - C. To monitor system performance
 - D. To manage user permissions
3. What is an image registry in the context of Docker?
 - A. A tool for monitoring container performance
 - B. A repository for storing Docker images
 - C. A local file system storage
 - D. A command-line interface for Docker
4. What does the command `'docker image prune -a'` do?
 - A. Removes all containers
 - B. Removes only the dangling images
 - C. Removes all unused images
 - D. Removes all stopped containers
5. How can you connect a container to an existing Docker network?
 - A. Using the command `docker link [container_id] [network_name]`
 - B. Using the command `docker network attach [network_name] [container_id]`
 - C. Using the command `docker network connect [network_name] [container_id]`
 - D. Using the command `docker network join [network_name] [container_id]`

- 6. What is the process of packaging an application with Docker known as?**
- A. Containerization**
 - B. Virtualization**
 - C. Serverization**
 - D. Deployment**
- 7. What does IEBNMT stand for in providing service to guests?**
- A. Introduce, Engage, Be polite, Never complain, Make connections, Thank**
 - B. Inform, Encourage, Brief, Never hesitate, Make amends, Thank**
 - C. Introduce, Engage, Be informative, Never say I don't know, Thank**
 - D. Inquire, Evaluate, Be nice, Notate, Thank**
- 8. Which of the following is NOT a nacho choice?**
- A. Veggie**
 - B. Chicken teriyaki**
 - C. Lobster fondue**
 - D. Banging shrimp**
- 9. What is the recommended first step for placing dirty dishes in the dish area?**
- A. Stack dishes without clearing food**
 - B. Clear all food into the garbage**
 - C. Spray dishes with cleaning solution**
 - D. Leave dishes on the table**
- 10. Which ingredients are found in the crab and lobster mac and cheese?**
- A. Fusilli noodles and cheddar cheese**
 - B. Spaghetti, crab, and cheddar cheese**
 - C. Cavatappi noodles, creamy cheese sauce, crab, and lobster**
 - D. Penne pasta, white sauce, and shrimp**

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. What does the `docker prune` command do?

- A. Creates a backup of Docker configurations
- B. Removes unused data from Docker**
- C. Updates Docker to the latest version
- D. Starts all stopped containers

The `docker prune` command is designed to remove unused data from Docker, which helps in reclaiming disk space and maintaining a cleaner working environment. It specifically targets items like stopped containers, unused networks, dangling images, and build cache that are no longer needed. By invoking this command, users can effectively clean up their Docker environment, ensuring that only necessary resources are retained. This functionality is crucial for managing disk usage, especially in setups where multiple containers and images are frequently created and destroyed. The ability to remove unused data helps streamline operations and ensures that the Docker environment remains efficient over time.

2. What is the primary function of the Docker API?

- A. To create new Docker containers
- B. To interact programmatically with the Docker daemon**
- C. To monitor system performance
- D. To manage user permissions

The Docker API's primary function is to interact programmatically with the Docker daemon. This API allows developers to communicate with the Docker daemon, which is the core component responsible for managing Docker containers, images, networks, and volumes. By using the Docker API, users can automate container management tasks, such as starting and stopping containers, building images, and querying the status of containers. This interaction is crucial for building tools and applications that need to manage Docker environments without manual intervention. The API provides a rich set of endpoints that cover all aspects of container management, enabling a high level of integration with various development processes and workflows. Other options, while related to Docker, do not reflect the primary function of the API itself. For example, creating new Docker containers is one of the many actions that can be performed through the API, but it does not capture the full scope of its capabilities. Similarly, monitoring system performance and managing user permissions involve different facets of system administration and security, which can utilize Docker APIs but do not define their core purpose.

3. What is an image registry in the context of Docker?

- A. A tool for monitoring container performance
- B. A repository for storing Docker images**
- C. A local file system storage
- D. A command-line interface for Docker

An image registry is fundamentally a repository designed specifically for storing and managing Docker images. This plays a crucial role in the development and deployment of containerized applications. When developers create a Docker image, they can push it to an image registry, making it available for sharing and deploying on different platforms or environments. Image registries serve as centralized locations where multiple images can be stored, organized, and versioned, allowing teams to collaborate efficiently. This helps streamline the process of retrieving images, ensuring that the correct version is used in various applications and services. The other options do not align with the definition of an image registry, as they either pertain to different aspects of Docker (such as performance monitoring, file storage, or command-line interfaces) that do not involve the management and storage of images. In summary, an image registry is an essential component in the Docker ecosystem, facilitating effective image management and sharing among developers and systems.

4. What does the command 'docker image prune -a' do?

- A. Removes all containers
- B. Removes only the dangling images
- C. Removes all unused images**
- D. Removes all stopped containers

The command 'docker image prune -a' is designed to help manage disk space by removing unused images from your Docker environment. When you use this command with the '-a' flag, it targets all images that are not currently being used by any containers, rather than only those that are dangling—meaning they are not associated with any tags. This is particularly useful for reclaiming space because, over time, unused images can accumulate and take up significant space on your system. By executing this command, you clean up your Docker images, ensuring that only the ones actively in use by running containers are retained. This functionality is a significant aspect of Docker's resource management, enabling users to maintain a more efficient and organized development environment.

5. How can you connect a container to an existing Docker network?

- A. Using the command `docker link [container_id] [network_name]`
- B. Using the command `docker network attach [network_name] [container_id]`
- C. Using the command `docker network connect [network_name] [container_id]`**
- D. Using the command `docker network join [network_name] [container_id]`

To connect a container to an existing Docker network, the appropriate command is "`docker network connect [network_name] [container_id]`". This command is specifically designed to add an already running container to an existing network. When a container is connected to a Docker network, it can communicate with other containers on that network using their container names or IP addresses. This connectivity is crucial for distributed applications where different components need to communicate with each other efficiently. The syntax of the command indicates that you specify the network name first, followed by the container ID. This clearly links the container to the specified network while maintaining the container's existing configurations and settings. Unlike other methods that may suggest outdated or incorrect practices, this command adheres to Docker's current networking features and best practices. Using this command allows you to easily integrate different services or applications running in different containers, fostering a flexible architecture for development and deployment.

6. What is the process of packaging an application with Docker known as?

- A. Containerization**
- B. Virtualization
- C. Serverization
- D. Deployment

The process of packaging an application with Docker is known as containerization. This method allows developers to bundle an application along with all its dependencies, libraries, and environment settings into a single unit called a container. Containers ensure that the application runs consistently across different computing environments, facilitating ease of deployment and scalability. Containerization leverages the operating system's features to create isolated environments, which makes it lightweight compared to traditional virtualization that relies on hypervisors to run separate operating systems for each application instance. This approach streamlines the development process, as developers can be confident that the application will behave the same way regardless of where it is deployed. By adopting containerization, teams can significantly enhance their workflow, improving both efficiency and productivity. The focus on packaging applications in containers has become a prevalent choice in modern development practices, especially in continuous integration and continuous deployment pipelines.

7. What does IEBNMT stand for in providing service to guests?

- A. Introduce, Engage, Be polite, Never complain, Make connections, Thank**
- B. Inform, Encourage, Brief, Never hesitate, Make amends, Thank**
- C. Introduce, Engage, Be informative, Never say I don't know, Thank**
- D. Inquire, Evaluate, Be nice, Notate, Thank**

The acronym IEBNMT stands for "Introduce, Engage, Be informative, Never say I don't know, Thank," which is a structured approach to providing excellent service to guests. Each component of this acronym emphasizes a key aspect of customer interaction that enhances the guest experience. - "Introduce" signifies the importance of a warm welcome and personal connection right at the beginning of the interaction. It sets a positive tone and indicates to the guest that they are valued. - "Engage" underscores the necessity of actively involving the guest in the conversation. This interaction fosters goodwill and makes the guest feel appreciated and listened to. - "Be informative" highlights the essential role of providing accurate and helpful information to guests. This reduces confusion and enhances their overall experience during their visit. - "Never say I don't know" suggests the importance of maintaining professionalism. Instead of admitting ignorance, the ideal response would be to seek the answer or direct the guest to someone who can help, which builds trust and confidence. - "Thank" is a crucial finale that reinforces appreciation for the guest's presence and engagement with our service. It leaves a lasting positive impression. Together, these elements create a comprehensive framework for delivering high-quality service and ensuring that guests feel valued and well cared for

8. Which of the following is NOT a nacho choice?

- A. Veggie**
- B. Chicken teriyaki**
- C. Lobster fondue**
- D. Banging shrimp**

The selection that is typically not considered a traditional nacho topping is chicken teriyaki. Nachos traditionally feature toppings like cheese, jalapeños, sour cream, beans, various meats (usually ground beef or chicken), and a variety of vegetables. While chicken can often be a nacho topping, the specific combination with teriyaki sauce is less common and strays from classic nacho flavors, which tend to favor more Mexican or Tex-Mex influences. Therefore, chicken teriyaki as a topping does not align with conventional nacho choices, making it the choice that stands out in this context. In contrast, options like veggie, lobster fondue, and banging shrimp reflect ingredients or preparations that, while unique, fit more closely within contemporary interpretations of nachos.

9. What is the recommended first step for placing dirty dishes in the dish area?

- A. Stack dishes without clearing food**
- B. Clear all food into the garbage**
- C. Spray dishes with cleaning solution**
- D. Leave dishes on the table**

The recommended first step for placing dirty dishes in the dish area is to clear all food into the garbage. This action is important because it helps maintain cleanliness and prevents food waste from contaminating the dishwashing area. By removing food remnants, you reduce the risk of cross-contamination and help ensure that the washing process is more effective. This practice also makes it easier to stack and rinse the dishes before they are cleaned, leading to a more efficient dishwashing routine. Proper sanitation and organization in the dish area are essential for maintaining a hygienic kitchen environment, and clearing food is a fundamental step in achieving that goal.

10. Which ingredients are found in the crab and lobster mac and cheese?

- A. Fusilli noodles and cheddar cheese**
- B. Spaghetti, crab, and cheddar cheese**
- C. Cavatappi noodles, creamy cheese sauce, crab, and lobster**
- D. Penne pasta, white sauce, and shrimp**

The inclusion of cavatappi noodles, creamy cheese sauce, crab, and lobster is characteristic of crab and lobster mac and cheese. Cavatappi, which is a spiral-shaped pasta, holds the creamy cheese sauce well, allowing for a rich and satisfying texture that complements the dish. The creamy cheese sauce enhances the flavors of both the crab and lobster, making it a luxurious and indulgent option. This combination brings together the sweet, delicate flavors of the seafood with the savory richness of the cheese sauce, creating a harmonious balance that is highly appealing in mac and cheese dishes. The presence of both crab and lobster adds a gourmet aspect that differentiates it from standard mac and cheese recipes, which typically use simpler ingredients. Other options lack the key elements of crab and lobster or do not reflect the traditional components associated with this gourmet dish.