
1. Do containers have to be on a specific operating system or platform?

- ☐ No, containers don't run on operating systems or platforms.
- ☐ Yes, containers only work on Linux-based operating systems and platforms.
- ☒ No, containers work on many types of operating systems and platforms.
- ☐ Yes, containers only work on Windows-based operating systems and platforms.



Correct

Correct! Containers can be used across multiple types of operating systems and platforms.

2. How are digital containers like shipping containers?

- ☐ Digital containers are not like shipping containers.
- ☐ Digital containers must be welded to computer motherboards, just like shipping containers must be welded.
- ☐ Both digital and shipping containers have to travel internationally.
- ☒ Container size and specs are standardized, which makes them easier to move around.



Correct

Correct! Digital containers make software portable so applications can run on multiple platforms.

3. What applications are a good fit for Docker?

- ☒ Applications that require flexible scaling and portability
- ☐ Applications with high performance requirements
- ☐ Applications with rich GUI features
- ☐ Applications that need a lot of security



Correct

Correct! Docker became popular with developers because of its simple architecture, high scalability, and easy portability.

4. What are three parts of Docker's underlying technology?

☒ The Go programming language

☒ **Correct**

Correct! The Go programming language is a part of Docker's underlying technology.

☒ Linux kernel features

☒ **Correct**

Correct! Linux kernel features are a part of Docker's underlying technology.

☒ Namespaces

☒ **Correct**

Correct! Namespaces are a part of Docker's underlying technology.

☐ GitHub

5. What Docker features create a container image?

☐ The image command and a Dockerfile

☐ The copy command and an existing image

☐ The run command and a Dockerfile

☒ The build command and a Dockerfile

☒ **Correct**

Correct! The build command is used with a Dockerfile to build a container image.

6. What are the steps used to create and run containers?

- ☐ Create a container image, use it to create a Dockerfile, and then use the Dockerfile to create a running container.
- ☐ Input the container image name and tag.
- ☐ Create a Dockerfile and use it with the pull command to create a running container.
- ☒ Create a Dockerfile, use it to create a container image, and then use the container image to create a running container.

✓ **Correct**

Correct! The proper sequence of steps to create and run containers is to create a Dockerfile, use it to create a container image, and then use the container image to create a running container.

7. What is the function of the Docker 'run' command?

- ☐ Stores images in a configured registry
- ☐ Lists all images, repositories, tags, and sizes
- ☐ Retrieves images from a configured registry
- ☒ Creates a container from an image

✓ **Correct**

Correct! The Docker 'run' command creates a container from an image.

8. What is a Docker container?

- ☐ A method of isolating communication
- ☒ A runnable instance of an image
- ☐ A persistent set of data that can be transferred
- ☐ A read-only template

✓ **Correct**

Correct! A Docker container is defined as a runnable instance of an image.

9. What are volumes and bind mounts used for in Docker?

- ☒ Persisting data
- ☐ Isolating communication
- ☐ Erasing data
- ☐ Connecting to external storage platforms

✓ **Correct**

Correct! Volumes and bind mounts are used to persist data in Docker.

10. What does the Docker client-server architecture provide?

- ☒ An application environment
- ☐ Cloud storage
- ☐ A communication channel
- ☐ Code checking

✓ **Correct**

Correct! The Docker client-server architecture provides a complete application environment.
