1. What are Docker containers?

Docker containers create an abstraction at the application layer and package applications together with all of their dependencies. This allows us to deploy applications quickly and reliably. Containers don’t require us to install a different operating system. Instead, they use the underlying system’s CPU and memory to perform tasks. This means that any containerized application can run on any platform regardless of the underlying operating system. We can also think of containers as runtime instances of Docker images.

2. What is a DockerFile?

A Dockerfile is a text file that contains all of the commands that we need to run to build a Docker image. Docker uses the instructions in the Dockerfile to automatically build images. We can use docker build to create automated builds that execute multiple command-line instructions in sequential order.

3. How do you create a Docker container from a Docker image?

To create a container from an image, we pull out the image that we want from the Docker repository and create a container. We can use the following command:

4. Can you use JSON instead of YAML for Docker Compose?

5. What is Docker Swarm?

Docker Swarm is a container orchestration tool that allows us to manage multiple containers across different host machines. With Swarm, we can turn multiple Docker hosts into a single host for easy monitoring and management.

6. If you want to use a base image and make modifications to it, how do you do that?

We can pull an image from Docker Hub onto our local system using the following Docker command:

7. How do you start, stop, and kill containers?

8. What platforms does Docker run on?

Docker runs on the following Linux distributions:

CentOS 6+

Gentoo

ArchLinux

CRUX 3.0+

openSUSE 12.3+

RHEL 6.5+

Fedora 19/20+

Ubuntu 12.04, 13.04

Docker can also be used in production with these cloud services:

Microsoft Azure, Google Compute Engine Amazon AWS EC2

9. Explain the Docker components.

10. What’s the difference between virtualization and containerization?

11. What is the functionality of a hypervisor?

12. How do you build a Dockerfile?

13. What command do you use to push a new image to the Docker Registry?

14. What is Docker Engine?

15. How do you access a running container?

16. How do you list all the running containers?

17. Describe the lifecycle of a Docker container.

18. What are Docker object labels?

19. How do you ensure that container 1 runs before container 2 while using Docker Compose?

20. What does the docker create command do?

The docker create command creates a writable container layer over a specified image and prepares that image for running the specified command.

20 more interview questions to explore

Explain Docker architecture.

What’s the difference between CMD and ENTRYPOINT?

What is the purpose of the volume parameter in a Docker run command?

Is it a good practice to run stateful applications on Docker?

What are Docker Namespaces?

Explain the implementation method of continuous integration and continuous deployment in Docker.

What is the process for stopping and restarting a Docker container?

How do you give your Docker image an image name?

What does the docker service command do?

Can you lose data when the container exits?

How do Jenkins and Docker work together?

How far do Docker containers scale?

Describe the differences between daemon logging and container logging.

Explain the purposes of up, run, and start commands of Docker compose.

Where are Docker volumes stored?

Explain the difference between Docker image and layer.

Can a paused container be removed from Docker?

How do you use the docker save and docker load commands?

What is the default Docker network driver? How can you change it when running a Docker image?

What does the docker system prune command do?