



Top 40 Docker interview questions (with answers)

Aug 06, 2021 - 8 min read



Erin Schaffer



Docker is a popular open-source software platform that simplifies the process of creating, managing, running, and distributing applications. It uses containers to package applications along with their dependencies. Docker dominates the market. Most of the top cloud (<https://www.educative.io/blog/roadmap-to-cloud-engineer-jobs>) and IT companies have adopted Docker to streamline their application development workflows. The demand for applicants with Docker experience is high.

Cracking your Docker interview is the key to landing one of these highly coveted roles. We've gathered the top 40 Docker interview questions to help you prepare for your Docker interview. This Docker tutorial includes both questions and answers. Let's get started!

We'll cover:

- Docker interview questions and answers
- 20 more interview questions to explore
- Next steps

Get hands-on with Docker for free

Master Docker fundamentals with Educative's 1-week free trial (<https://www.educative.io/trial>).

Your Privacy

We store cookies on your device to provide necessary website functionality, improve your experience and analyze traffic. To learn more, please review our [Cookie Policy](#) ([/cookie-policy](#)).

Got It

Docker interview questions and answers

1. What are Docker containers?

Docker containers (<https://www.educative.io/blog/beginners-guide-to-docker>) 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:

```
$ docker run -it -d <image_name>
```

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

Yes, we can use a JSON file instead of a YAML (<https://www.educative.io/blog/yaml-tutorial>) file for the Docker Compose (<https://www.educative.io/blog/docker-compose-tutorial>) file. To use JSON, we need to specify the filename like this:

Your Privacy

```
$ docker-compose -f docker-compose.json up
```

We store cookies on your device to provide necessary website functionality, improve your experience and analyze traffic. To learn more, please review our [Cookie Policy](#) ([cookie-policy](#)).

5. What is Docker Swarm? Got It

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:

```
$ docker pull <image_name>
```

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

To start a Docker container, use the following command:

```
$ docker start <container_id>
```

To stop a Docker container, use the following command:

```
$ docker stop <container_id>
```

To kill a Docker container, use the following command:

```
$ docker kill <container_id>
```

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 (<https://www.educative.io/blog/microservices-in-azure>)
- Google Compute Engine
- Amazon AWS EC2
- Amazon AWS ECS
- Rackspace

We store cookies on your device to provide necessary website functionality, improve your experience and analyze traffic. To learn more, please review our [Cookie Policy](#) ([/cookie-policy](#)).

Got It

Tip: We always recommend engaging in some company research prior to your interview. To prepare for this particular question, find out how the company uses Docker and include the platform they use in your answer.

9. Explain the Docker components.

The three architectural components include Docker Client, Host, and Registry.

- **Docker Client:** This component executes build and run operations to communicate with the Docker Host.
- **Docker Host:** This component holds the Docker Daemon, Docker images, and Docker containers. The daemon sets up a connection to the Docker Registry.
- **Docker Registry:** This component stores Docker images. It can be a public registry, such as Docker Hub or Docker Cloud, or a private registry.

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

Virtualization

Virtualization helps us run and host multiple operating systems on a single physical server. In virtualization, hypervisors give a virtual machine to the guest operating system. The VMs form an abstraction of the hardware layer so each VM on the host can act as a physical machine.

Containerization

Containerization provides us with an isolated environment for running our applications. We can deploy multiple applications using the same operating system on a single server or VM. Containers form an abstraction of the application layer, so each container represents a different application.

Keep learning Docker for free.

Get started with Docker for free with our 1-week Educative Unlimited Trial (<https://www.educative.io/trial>). Educative's text-based learning paths are easy to skim and feature live coding environments, making learning quick and efficient.

DevOps for Developers (<https://www.educative.io/path/devops-for-developers>)

We store cookies on your device to provide necessary website functionality, improve your experience and analyze traffic. To learn more, please review our [Cookie Policy](#) ([/cookie-policy](#)).

Got It

11. What is the functionality of a hypervisor?

A hypervisor, or virtual machine monitor, is software that helps us create and run virtual machines. It enables us to use a single host computer to support multiple guest virtual machines. It does this by dividing the system resources of the host and allocating them to the installed guest environments. Multiple operating systems can be installed on a single host operating system. There are two kinds of hypervisors:

- **Native:** Native hypervisors, or bare-metal hypervisors, run directly on the underlying host system. It gives us direct access to the hardware of the host system and doesn't require a base server operating system.
- **Hosted:** Hosted hypervisors use the underlying host operating system.

12. How do you build a Dockerfile?

In order to create an image with our outlined specifications, we need to build a Dockerfile. To build a Dockerfile, we can use the `docker build` command:

```
$ docker build <path to dockerfile>
```

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

To push a new image to the Docker Registry, we can use the `docker push` command:

```
$ docker push myorg/img
```

14. What is Docker Engine?

Docker Engine is an open-source containerization technology that we can use to build and containerize our applications. Docker Engine is supported by the following components:

- Docker Engine REST API
- Docker Command-Line Interface (CLI)
- Docker Daemon

15. How do you access a running container?

Your Privacy

We store cookies on your device to provide necessary website functionality, improve your experience and analyze traffic. To learn more, please review our [Cookie Policy](#) ([/cookie-policy](#)).

To access a running container, we can use the following command:

Got It

```
$ docker exec -it <container_id> bash
```

16. How do you list all the running containers?

To list all of the running containers, we can use the following command:

```
$ docker ps
```

17. Describe the lifecycle of a Docker container.

Docker containers go through the following stages:

- Create a container
- Run the container
- Pause the container (optional)
- Un-pause the container (optional)
- Start the container
- Stop the container
- Restart the container
- Kill the container
- Destroy the container

18. What are Docker object labels?

Docker object labels are key-value pairs that are stored as strings. They enable us to add metadata to Docker objects such as containers, networks, local daemons, images, Swarm nodes, and services.

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

Docker Compose doesn't wait for containers to be ready before moving forward with the next container. In order to control our order of execution, we can use the "depends on" condition, `depends_on`. Here's an example of it being used in a `docker-compose.yml` file:

```
version: "2.4"
```

```
services:
```

```
  backend:
```

```
    build: .
```

```
    depends_on:
```

```
      - db
```

```
  db:
```

```
    image: postgres
```

Your Privacy

We store cookies on your device to provide necessary website functionality, improve your experience and analyze traffic. To learn more, please review our [Cookie Policy \(/cookie-policy/\)](#).

Got It

The `docker-compose up` command will start and run the services in the dependency order that we specify.

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

- 21. Explain Docker architecture.
- 22. What's the difference between CMD and ENTRYPOINT?
- 23. What is the purpose of the volume parameter in a Docker run command?
- 24. Is it a good practice to run stateful applications on Docker?
- 25. What are Docker Namespaces?
- 26. Explain the implementation method of continuous integration and continuous deployment in Docker.
- 27. What is the process for stopping and restarting a Docker container?
- 28. How do you give your Docker image an image name?
- 29. What does the `docker service` command do?
- 30. Can you lose data when the container exits?
- 31. How do Jenkins and Docker work together?
- 32. How far do Docker containers scale?
- 33. Describe the differences between daemon logging and container logging.
- 34. Explain the purposes of `up`, `run`, and `start` commands of Docker compose.
- 35. Where are Docker volumes stored?
- 36. Explain the difference between Docker image and layer.
- 37. Can a paused container be removed from Docker?
- 38. How do you use the `docker save` and `docker load` commands?
- 39. What is the default Docker network driver? How can you change it when running a Docker image?
- 40. What does the `docker system prune` command do?

We store cookies on your device to provide necessary website functionality, improve your experience and analyze traffic. To learn more, please review our [Cookie Policy](#) ([Cookie Policy](#)).

system prune command

Got It

Next steps

Congrats! You made it to the end. Preparing for your Docker interview will take time, so be patient with the process. Be prepared to spend a lot of time studying and preparing. There's still more to learn about Docker. Some recommended topics to cover next include:

- Docker with common development profiles
- Docker security
- Private registries

To get started learning these concepts and a lot more, check out Educative's learning path **DevOps for Developers** (<https://www.educative.io/path/devops-for-developers>). In this curated learning path, you'll get hands-on practice with Docker and Kubernetes. By the end, you'll have cutting-edge skills and hands-on experience so you can excel in your DevOps role.

Happy learning!

Continue reading about interview prep

- 50 Python interview questions and answers (<https://www.educative.io/blog/python-interview-questions>)
- Top 50 React Native interview questions (with answers) (<https://www.educative.io/blog/top-react-native-interview-questions>)
- Top 5 hardest coding interview questions from recent FAANG interviews (<https://www.educative.io/blog/hardest-coding-interview-questions>)



WRITTEN BY

Erin
Schaffer

Join a community of more than 1.3 million readers. A free, bi-monthly email with a roundup of Educative's top articles and coding tips.

Your Privacy

Full Name

We store cookies on your device to provide necessary website functionality, improve your experience and analyze traffic. To learn more, please review our [Cookie Policy \(/cookie-policy\)](#).

E-mail

Subscribe

Got It



PRODUCTS

Educative Learning

(/individual-learner)

Educative

Onboarding

(/onboarding)

Educative Skill

Assessments

(/assessments)

PRICING

For Enterprise

(/enterprise-pricing)

For Individuals

(/unlimited)

LEGAL

Privacy Policy

(/privacy)

-Terms of Service

(/terms)

Business Terms of

Service

(/enterprise-terms)

MORE

Course Catalog

(/explore)

Early Access

Courses

(/explore/early-access)

Free Trials

(/h2c-trial)

Earn Referral Credits

(/refer-a-friend)

CodingInterview.com

We store

traffic. To

(<https://codinginterview.com/>?)

Prntsc=10449898.a424d9567829dbd2703b23d3645e216f.1654607814048.1654607814048.1654607814048.1&__hssc=10449898

[illegible]

(/press)

Contact Us

(/contactUs)



ook.com/educativeinc) (/linkedin.com/company/educative- (/twitter.com/educativeinc) (/www.youtube.com/channel/UCT_8FqzTlr2Q1BOtvX_DPPw/? sub_confirmation=1) (/educativesessions.podt



Copyright ©2022 Educative, Inc. All rights reserved.

Your Privacy

We store cookies on your device to provide necessary website functionality, improve your experience and analyze traffic. To learn more, please review our [Cookie Policy \(/cookie-policy\)](#).

Got It