



Courses

[Free Courses](#) [Interview Questions](#) [Tutorials](#) [Community](#)

Home / Interview Question / Top Docker Interview Questions and Answers

Top Docker Interview Questions and Answers

By Naveen 2.1 K Views 21 min read Updated on April 22, 2022

Go through the top industry-selected Docker interview questions that will help you prepare for your Docker interview. They include unique features of Docker, 'what is a Docker image?', Docker Hub, Docker Swarm, Docker Compose, how to start and stop a Docker container, and so on. Learn Docker from Intellipaat's **DevOps Course** and fast-track your career!

Become a Certified Professional

Categories	
Automation	3
Big Data	12
Business Intelligence	21
Cloud Computing	19
Cyber Security	2
Data Science	11
Database	6
Digital Marketing	2
Mobile Development	2
No-SQL	5
Programming	16
Project Management	5
Salesforce	3
Testing	4
Website Development	6

Top Answers to Docker Interview Questions

With years passing by, Docker has been showing exponential growth in terms of the rise in job openings. Professionals who have mastered the skills in Docker are in high demand today. The Docker Interview Questions and Answers blog is mainly created to give you an idea of the type of questions that are asked during Docker job interviews. So, take advantage of the latest and high-paying Docker job opportunities by preparing for your interview with these interview questions:

- [Q1. What is Docker?](#)
- [Q2. Define Containerization.](#)
- [Q3. What is the benefit of using a Docker over a Hypervisor?](#)
- [Q4. What are the unique features of Docker over other containerization technologies?](#)
- [Q5. What is a Docker image?](#)
- [Q6. What is a Docker container?](#)
- [Q7. What is a Docker Hub?](#)
- [Q8. What is a Docker Swarm?](#)
- [Q9. What is the use of a Dockerfile?](#)
- [Q10. What is Docker Compose?](#)

These Docker interview questions are broadly divided into major categories such as:

- [1. Basic](#)
- [2. Intermediate](#)
- [3. Advanced](#)

Watch this Docker Tutorial for Beginners to get a kick start:



Basic Interview Questions

1. What is Docker?

Criteria	Docker	Virtual Machines
Use of OS	All containers share the host OS	Each VM runs on its own OS

Startup time	Very fast	Slow
Isolation	Process-level isolation	Full isolation
Security	Low	High

We can define Docker as a containerization platform that combines all our applications in a package so that we have all the dependencies to run our applications in any environment. This means, our application will run seamlessly on any environment, and this makes it easy for having a product-ready application. What Docker does is wrap the software needed in a file system that has everything for running the code, providing the runtime and all the necessary libraries and system tools. Containerization technology like Docker will share the same operating system kernel with the machine, and due to this it is extremely fast. This means that we have to run Docker only at the beginning and after that, since our OS is already running, we will have a smooth and seamless process.

2. Define Containerization.

Containerization is a form of virtualization through which applications are run in containers (isolated user spaces) all using a shared OS. It packs or encapsulates software code and all its dependencies for it to run in a consistent and uniform manner on any infrastructure.

Have you checked out the [Docker Cheat Sheet](#) yet?

3. What is the benefit of using a Docker over a Hypervisor?

Though Docker and Hypervisor might do the same job overall, there are many differences between them in terms of how they work. Docker can be thought of as lightweight since it uses very few resources and also the host kernel rather than creating it like a Hypervisor.

4. What are the unique features of Docker over other containerization technologies?

Some of the most important and unique features of Docker are as follows:

- We can run our Docker container either on our PC or on our enterprise IT system.
- Along with the Docker Hub, which is a repository of all containers, we can deploy and download all our applications from a central location.
- We can even share our applications with the containers that we create.

Prepare yourself for the DevOps interview by learning from these [Top DevOps Interview Questions and Answers!](#)

5. What is a Docker image?

A Docker image helps in creating Docker containers. We can create the Docker image with the build command; due to this, it creates a container that starts when it begins to run. All the Docker images are stored in the Docker registry such as the public Docker registry. These have minimal amounts of layers within the image so that there is a minimum amount of data on the network.

Get 100% Hike!

Master Most in Demand Skills Now !

Email Address

▼

Phone Number

Submit

6. What is a Docker container?

It is a comprehensive set of applications including all its dependencies which share the same OS kernel, along with the other containers running in separate processes within the operating system in a user space. Docker is not tied to any IT infrastructure, and thus it can run on any computer system or on the cloud. We can create a Docker container using Docker images and then run it, or we can use the images that are already created in the Docker Hub. To simplify things, let's say that the Docker containers are just runtime instances of the Docker image.

7. What is a Docker Hub?

We can think of Docker Hub as a cloud registry that lets us link the code repositories, create the images, and test them. We can also store our pushed images, or we can link to the Docker Cloud, so that the images can be deployed to the host. We have a centralized container image discovery resource that can be used for the collaboration of our teams, automating the workflow and distribution, and changing management by creating the development pipeline.

Want to get certified in DevOps? Check out our [DevOps Certification](#) blog!

8. What is a Docker Swarm?

We can think of a Docker Swarm as the way of orchestrating the Docker containers. We will be able to implement Dockers in a cluster. We can convert our Docker pools into a single Docker Swarm for easy management and monitoring.

Go through the [Best DevOps Course in New York](#) to get a clear understanding of DevOps!

9. What is the use of a Dockerfile?

A Dockerfile is a set of specific instructions that we need to pass on to Docker so that the images can be built. We can think of the Dockerfile as a text document which has all the commands that are needed for creating a Docker image. We can create an automated build that lets us execute multiple command lines one after the other.

Learn more about DevOps in this [DevOps Training in Sydney](#) to get ahead in your career!

10. What is Docker Compose?

Docker Compose defines and runs multi-container Docker applications. With Compose, a YAML file is used to configure an application's services. All the services from the configuration can be created and started with a single command.

Interested in becoming a DevOps expert? Learn more from this [DevOps Course in Toronto](#)!

11. Why use Docker?

Following are the reasons why one should use Docker:

- It allows the use of system resources more efficiently
- Software delivery cycles are faster with it
- Application portability is possible and easy
- It is great for the microservices architecture

12. What are the drawbacks of Docker?

Docker has a few drawbacks as listed below:

- No storage option
- Poor monitoring
- Unable to automatically reschedule inactive nodes
- Has a complicated automatic horizontal scaling setup

13. What is Docker Engine?

Docker Engine is an open source containerization technology that facilitates the development, assembling, shipping, and

Docker Engine is an open-source containerization technology that facilitates the development, assembling, snipping, and running of applications with the help of the following components:

- Docker Daemon
- Docker Engine REST API
- Docker CLI

Career Transition

14. What are registries?

Docker registries provide locations for storing and downloading images. There are two types of registries

- Public registry
- Private registry

Public registries include Docker Hub and Docker Cloud.

15. What are Docker namespaces?

When a container is run, Docker creates a set of isolated workspaces for the container called namespaces.

Intermediate Interview Questions

16. How to use Docker?

The Docker command syntax looks like this:

```
docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
```

To run a container, we must incorporate the image on which it is based:

```
docker run [docker_image]
```

We can run containers if the Docker images are locally stored. If they are not, the software will take it from the online registry.

Docker can be used to run a container:

- Under a specific name
- Interactively
- And publish container ports
- In the background in the detached mode
- And mount host volumes
- And remove it once the process is complete

17. Is it possible to use JSON instead of YAML for Docker Compose?

We can use JSON instead of YAML for a Docker Compose file. When we are using the JSON file for composing, we have to specify the filename with the following command:

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

18. What is the process for creating a Docker container?

We can use any specific Docker image for creating a Docker container using the below command:

```
docker run -t -i command name
```

This command not only creates the container but also starts it for us. If we want to check whether the Docker container has been created or not, then we need to have the following command that will list all the Docker containers, along with the host on which the Docker containers run:

```
docker ps -a
```

Interested in getting an industry-recognized certification in DevOps? Enroll in Intellipaat's [DevOps Course in Bangalore](#) now!

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

To stop a Docker container, we need to use the following command:

```
docker stop CONTAINER_ID
```

To restart a Docker container, we need to use the following command:

```
docker restart CONTAINER_ID
```

Prepare yourself for the DevOps Certification with this comprehensive [DevOps Training in Hyderabad!](#)

20. How to create a Docker image?

A Docker image can be built using the following command:

```
$ docker build .
```

Courses you may like

21. How to stop a Docker container?

We can use the following to stop one or more running Docker containers:

```
docker container stop [OPTIONS] CONTAINER [CONTAINER...]
```

22. How to remove Docker images?

Use **docker images** with the **-a** flag to get the image IDs for removal. Then, pass their IDs or tags to

```
docker rmi:
```

List:

```
docker images -a
```

Remove:

```
docker rmi Image Image
```

23. Where are Docker images stored?

It depends on which system the Docker is running and the Docker storage driver is being used.

For example, on Windows, Docker images are stored by default in:

```
C:\ProgramData\DockerDesktop
```

On a Mac, Docker images are stored by default in:

```
~/Library/Containers/com.docker.docker/Data/vms/0/
```

24. How to run a Docker image?

The 'docker run' command runs an image inside the container. It will require the image name.

```
docker run [docker_image]
```

25. How to start a Docker container?

The following command starts a Docker container:

```
docker container start [OPTIONS] CONTAINER [CONTAINER...]
```

26. How to use Docker Compose?

Docker Compose typically includes a three-step process:

1. Using a **dockerfile** to define the app's environment to facilitate reproduction anywhere
2. Defining the app services in **docker-compose.yml** so that they can run in an isolated environment together
3. Running **docker-compose up**

27. What command should be run to view all the running Docker containers?

To view all the running containers in Docker, we can use the following:

```
$ docker ps
```

28. What is Docker daemon?

Docker daemon is a service that manages Docker containers, images, storage volumes, and the network. It constantly listens to Docker API requests and processes them. A daemon can communicate with other daemons as well for the management of Docker services.

29. Name and explain the states of a Docker container.

- **Created:** We see this Docker container state when a container is newly created.
- **Restarting:** When the Docker container is restarted due to any issues, this state is observed.
- **Running:** It is the main state for the container after it has started.
- **Paused:** When a running Docker container is temporarily stopped via docker pause, this is the status that we will see.
- **Exited:** If a container has stopped due to some issue or stopped manually, this will be the state of the container.
- **Dead:** When the daemon has tried but failed to stop a container (mostly because of a busy device or resource), this state will be seen.

30. What is Docker Hub?

Docker Hub helps with linking to code repositories. This cloud-based registry enables the building, testing, and storing of images in Docker Cloud. Images can also be deployed to the host with it.

31. Define Virtualization.

Virtualization is the process of logically dividing mainframes to enable more than one application to run at a time.

32. What is a Hypervisor?

A hypervisor helps in the creation of a virtual environment, in which the guest virtual machines run. It manages the guest systems and checks the required resource allocations to the guests.

33. What are Docker object labels?

Docker object labels help us add metadata to Docker objects, including containers, images, Swarm nodes, network, volumes, and services.

34. What are the steps in a Docker container life cycle?

- Build
- Pull
- Run

35. Name the three components of the Docker architecture.

- Client
- Docker host
- Registry

Advanced Interview Questions

36. How do you scale your Docker containers?

Docker containers can be scaled to any level, starting from a few hundreds to even thousands or millions of containers. The only condition is that the containers need the memory and the OS all the time, and there should not be a constraint on these when the Docker is getting scaled.

37. How does communication happen between Docker client and Docker daemon?

The communication between Docker client and Docker daemon happens with the help of the combination of TCP, Rest API, and Socket.IO.

38. Explain the implementation method of continuous integration (CI) and continuous

deployment (CD) in Docker.

- Run [Jenkins](#) on Docker
- Using **docker-compose**, run integration tests in Jenkins

39. How to run a Docker container?

The Docker run command manages the running of containers in Docker.

Running a container under a specific name:

The command for running a container under a specific name is:

```
docker container run --name [container_name] [docker_image]
```

Running a container in the background in the detached mode:

The command for running a container in the background is:

```
docker container run -d [docker_image]
```

Running a container interactively:

The following command is run for running a container interactively:

```
docker container run -it [docker_image] /bin/bash
```

Running a container and publishing container ports:

We have to include **-p** to the **docker run** command, along with the following:

```
-p [host_ip]:[host_port]:[container_port]
```

Here, **host_ip** is optional. It is not mandatory to specify this while we run the command.

Running a container and mounting host volumes:

The **docker container run** command looks like this:

```
docker container run -v [/host/volume/location]:[/container/storage] [docker_image]
```

40. Tell us how you have used Docker in your past position.

This is a question wherein we could bring upon our whole experience with Docker and any other Container technologies we have used prior to Docker. We could also explain the ease that this technology has brought in the automation of the development-to-production life cycle management. We can also discuss any other integrations that we might have worked, along with Docker, such as Puppet, Chef, or even the most popular of all technologies—Jenkins. If we do not have any experience with Docker but we have it with similar tools from this space, we could convey the same and also show our interest in learning this leading containerization technology.

If you have any doubts or queries related to DevOps, get them clarified from DevOps experts on our [DevOps Community](#)!

[Next](#)

Name	Date	
AWS Certification	2022-05-28 2022-05-29 (Sat-Sun) Weekend batch	View Details
AWS Certification	2022-06-04 2022-06-05 (Sat-Sun) Weekend batch	View Details
AWS Certification	2022-06-11 2022-06-12 (Sat-Sun) Weekend batch	View Details

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Post Comment

Find Best DevOps Training in Other Regions

- [Bangalore](#)[Melbourne](#)[Chicago](#)[Hyderabad](#)[San Francisco](#)[London](#)[New York](#)[Toronto](#)[India](#)[Chennai](#)[Delhi](#)
- [Gurgaon](#)[Kolkata](#)[Mumbai](#)[Pune](#)[Singapore](#)[Dubai](#)[Houston](#)[Jersey City](#)[Los Angeles](#)[Noida](#)[Ahmedabad](#)
- [Coimbatore](#)[Sydney](#)[Dallas](#)[Atlanta](#)[Michigan](#)[Manchester](#)[Montreal](#)