Docker Interview Questions and Answers

Introduction

Docker interview questions and answers commonly cover containerization concepts, Docker architecture, and the differences between containers and virtual machines. Candidates are also asked about Docker images, volumes, networking, and real-world deployment scenarios. These questions evaluate hands-on skills in container management and DevOps practices. New to this topic? Build a rock-solid foundation first with our Docker tutorial for beginners before diving into these advanced Docker interview questions.

List of Docker Interview Questions for Freshers

1. What is virtualization?
2. What is containerization?
3. What is Docker?

4. What features does Docker offer?
5. Describe the Docker components.
6. Describe a Docker container's state.
7. What features does a hypervisor offer?
8. Define Docker Hub
9. Describe Docker Swarm.
10. What is Docker Namespace?

Don't just learn, master the craft. Download our Docker course syllabus to see our week-by-week breakdown of industry-essential skills.

Docker Interview Questions and Answers for Freshers

1. What is virtualization?

The process of turning objects like compute storage, servers, applications, etc. into a software-based virtual version is called virtualization. One physical hardware system serves as the basis for the creation of these virtual versions or environments.1

2. What is containerization?

All of the configuration files and dependencies for an application are bundled and wrapped together when it is developed and deployed. We refer to this bundle as a container.

Now that all the dependencies and libraries are bundled together, the container is deployed, providing a bug-free environment when you want to execute the program on a different machine. The two most well-known containerization environments are Kubernetes and Docker.

3. What is Docker?

Docker is a framework for containerization that enables the packaging of an application and all of its dependencies into a single, easily deployable container that can run on any Docker-compatible host.

This facilitates the development, testing, and deployment of programs in many environments. It isolates processes and offers a lightweight, portable application deployment solution using container technology.

4. What features does Docker offer?

With Docker, you may use containerization to achieve consistent deployment, efficient resource usage through shared kernel utilization, and smooth environment portability. Through container isolation that facilitates versioning and automated builds, it improves security. For quicker application development and deployment, it provides an abundance of pre-built images.

5. Describe the Docker components.

The following are the components of Docker:

- ★ The runtime that puts containers to use is called *Docker Engine*.
- ★ *Docker images* are executable packages that include the program and all of its dependencies. They are readable, lightweight templates.
- ★ Applications or instances of Docker images can be run in standardized, enclosed environments called *Docker containers*.
- ★ A tool for creating and executing multi-container Docker applications is called *Docker Compose*.

6. Describe a Docker container's state.

A Docker container's state has a direct impact on its runtime properties and how it communicates with the operating system underneath. One of these three states will characterize a Docker container:

- ★ When a Docker container is actively executing, it is running.
- ★ A container that is in the paused state has been put on hold for the time being.
- ★ When the container is not in use, it will be in a paused state.

7. What features does a hypervisor offer?

The hypervisor is a virtualization tool that facilitates the operation of several operating systems (Guest OS) on just one physical host system by isolating and managing the resources of the virtual machines (VMs).

8. Define Docker Hub

Docker containers are made using Docker images. These Docker images must be stored in a registry. This is the Docker Hub registry. To create personalized images and containers, users can select images from Docker Hub. At the moment, the world's biggest public image container repository is the Docker Hub.

9. Describe Docker Swarm.

The native clustering for Docker is called Docker Swarm. It creates a single virtual Docker host out of a pool of Docker hosts. Any program that already interacts with a Docker daemon can use Docker Swarm to transparently scale to several hosts since it supports the standard Docker API.

10. What is Docker Namespace?

One of the key concepts of containers as a component of Linux is a namespace. In containers, the namespace provides an extra degree of isolation. To remain portable and not interfere with the underlying host system, Docker offers a variety of namespaces. Docker supports a few namespace types: PID, Mount, IPC, User, and Network.

Enroll for our best **Docker training in Chennai** to learn further.

List of Docker Interview Questions for Experienced

1. What is a Docker container's lifecycle?
2. How can you find out the version of the Docker client and Docker server?
3. How can the number of containers that are paused, stopped, and running be obtained?
4. How can you access the Docker repository account?
5. Write code that starts, stops, and kills a container.
6. What distinguishes Docker from other containerization techniques?
7. Which operating systems support Docker?
8. Is it preferable to use the rm command to remove the container immediately or to stop it first and then remove it?
9. When transferring your Docker Compose file to production, what modifications should be made?

10. How is a Dockerfile constructed?

Skip the trial and error with our expert-led Docker challenges and solutions and master these fixes in hours, not days.

Docker Technical Interview Questions and Answers for Experienced

1. What is a Docker container's lifecycle?

A Docker container's lifespan looks like this:

- ➔ Establish a container
- ➔ Launch the container
- ➔ (Optional) Pause the container
- ➔ Unpause the container (if desired).
- ➔ Launch the container.
- ➔ Put an end to the container
- ➔ Start the container again.
- ➔ Take out the container or destroy it.

2. How can you find out the version of the Docker client and Docker server?

We may learn more about the Docker client and server versions by running the following command:

$ docker version

3. How can the number of containers that are paused, stopped, and running be obtained?

To obtain comprehensive details about the Docker installed on your system, we can execute the following command.

$ docker info

4. How can you access the Docker repository account?

To log in to hub.docker.com, execute the following command:

$ docker login

We will be asked to enter your username and password; after we do, we'll be logged in.

5. Write code that starts, stops, and kills a container.

To launch a Docker container, use the following command:

$ docker start <container_id>

and the subsequent steps to terminate an active container:

$ docker stop <container_id>

Use the following command to kill a container:

$ docker kill <container_id>

6. What distinguishes Docker from other containerization techniques?

Deploying Docker containers on any cloud platform is quite simple. When compared to other technologies, it can run more applications on the same hardware, facilitate the rapid creation of ready-to-run containerized applications by developers, and simplify the process of managing and deploying applications. Even sharing containers with your apps is possible.

7. Which operating systems support Docker?

Docker operates on multiple Linux operating systems:

- ➢ Ubuntu 12.04, 13.04 et al
- ➢ Fedora 19/20+
- ➢ RHEL 6.5+
- ➢ CentOS 6+
- ➢ Gentoo
- ➢ ArchLinux
- ➢ openSUSE 12.3+
- ➢ CRUX 3.0+

Additionally, it can be used in production with the following services on cloud platforms:

- ★ Amazon EC2
- ★ ECS Amazon
- ★ Google Compute Engine
- ★ Microsoft Azure
- ★ Rackspace

8. Is it preferable to use the rm command to remove the container immediately or to stop it first and then remove it?

It is usually preferable to use the remove command to end the container after stopping it.

$ docker stop <coontainer_id>

$ docker rm -f <container_id>

Sending the SIG_HUP signal to recipients can be accomplished by stopping the container and then deleting it. This will guarantee that each container has adequate time to finish its responsibilities and clean up. This approach is regarded as best practice since it prevents unintentional mistakes.

9. When transferring your Docker Compose file to production, what modifications should be made?

To prepare your application for production environment migration, apply the following modifications to your compose file:

- ★ To prevent changes to the code from occurring outside of the container, remove volume bindings.
- ★ Binding to many hosts' ports.
- ★ Give a policy for restarts.
- ★ Add more services, such as a log aggregator.

10. How is a Dockerfile constructed?

After writing a Dockerfile, it must be built to produce an image that meets those requirements.

The command to create a Dockerfile is as follows:

$ docker build <path to docker file>

The next query would be: should you use the full path or just ".dockerfile_name"?

If the Dockerfile lives elsewhere, use the whole path; otherwise, use ".dockerfile_name" if it exists in the same file directory.

Conclusion

After preparing these **Docker interview questions and answers**, you will gain a solid understanding of Docker, and you may proceed with confidence to share your learnings about efficient DevOps procedures. Enroll in our software training institute in Chennai for comprehensive learning on Docker with hands-on exposure.