# Docker Interview questions and answers

**Q1. What Is Docker?**

Docker is an open source containerization platform which is used for easy deployment of applications. Its enables developer to package the application into container,

**Q2. How are containers different from VM?**

Containers are very light weight in nature because they don't have a complete operating system. Containers are lightweight because they share the host OS kernel and only include necessary dependencies, while VMs are heavier as they run their own complete operating system. Containers offer faster startup times, better resource utilisation, and easier portability compared to VMs.

**Q3. What is Docker Lifecycle?**

In my case I will start writing dockerfile once I feel dockerfile is completed then I will execute dockerfile to create images by using docker run command. Once images are created I will run docker build command to create a container and finally push the image to the registry.

**Q4. What are the different docker components?**

Docker consists of several key components:

**Docker Engine:** The core component of Docker, responsible for building, running, and managing containers. It includes the Docker daemon, which listens for Docker API requests, and the Docker CLI (Command Line Interface), which allows users to interact with Docker through commands.

**Docker Images:** Immutable files that contain the application code, libraries, dependencies, and other files needed to run a containerized application. Images are used as the basis for creating Docker containers.

**Docker Containers:** Runnable instances of Docker images. Containers encapsulate the application and its dependencies, providing a lightweight, isolated environment for running applications. They can be started, stopped, and managed using Docker commands.

**Docker Registry:** A centralised repository for storing and distributing Docker images. The Docker Hub is a public registry provided by Docker, but organisations can also set up their own private registries to store proprietary or sensitive images.

**Q5. What Is the difference between Docker COPY and Docker ADD?**

Docker COPY is used to copy the files from host to container. Whereas Docker ADD is used to copy the files from URL and it will automatically unzip the .tar.gz files.

**Q6. What is the difference between CMD and ENTRYPOINTS in Docker?**

**CMD**: Specifies the default command to run when the container starts. It can be overridden by providing a command when running the container.

**ENTRYPOINT**: Specifies the command to run when the container starts. It cannot be overridden, but additional arguments can be passed to it when running the container.

In clearer terms, CMD sets the default command that can be easily overridden, while ENTRYPOINT sets the main command that cannot be overridden but can accept additional arguments.

**Q7. What are the networking types in Docker and What is the default?**

Default networking in Docker is Bridge.

However you can change the default type and configure one of them

- Bridge
- Overlay
- Host
- MacVlan

**Q8. Can you explain how to isolate networking between containers?**

let's take example I have created 2 containers whose name is login container and logout container with the default networking i.e bridge networking while creating 3rd container whose name is payment container Before I will create networking by using docker network create custom network by using **docker network create <name_of_networking>** and then run container with **— network=<name_of_networking>** so that my payment container will not talk to container login and container logout but it will talk to host or ec2 instance via **custom network i.e <name_of_networking>.**

**Q9. What is multistage Build in Docker?**

Multistage builds in Docker allow you to create more efficient Docker images by using multiple build stages within a single Dockerfile allowing you to copy artefacts from one stage to another. This approach helps reduce the size of the final Docker image and improves build efficiency.

**Q10. What is destro less images in Docker?**

- Destro less images contain only application and its runtime dependencies with minimum operating system libraries.
- They do not contain any package managers ad shell or any programs you would expect to find in standard Linux distribution.
- They are very small and lightweight in nature.

**Q11. Real time challenges with Docker?**

- Docker is a single daemon process. Which can cause a single point of failure, If the Docker Daemon goes down for some reason all the applications are down.
- Docker Daemon runs as a root user. Which is a security threat. Any process running as a root can have adverse effects. When it is compromised for security reasons, it can impact other applications or containers on the host.
- Resource Constraints: If you’re running too many containers on a single host, you may experience issues with resource constraints. This can result in slow performance or crashes.

**Q12. What steps would you take to secure containers ?**

1. Use Distroless or Images with not too many packages as your final image in multi stage build, so that there is less chance of CVE or security issues.
2. Ensure that the networking is configured properly. This is one of the most common reasons for security issues. If required, configure custom bridge networks and assign them to isolate containers.
3. Use utilities like Sync to scan your container images.