

## Basic Docker Questions:

### 1. What is Docker?

Docker is an open-source platform that uses containerization to package applications and their dependencies into isolated environments, ensuring they run consistently across different systems.

### 2. What is a Docker container?

A Docker container is a lightweight, standalone, and executable package that includes everything needed to run an application — code, runtime, libraries, and system tools.

### 3. What is the difference between Docker image and container?

- **Docker image:** A blueprint or template containing the app and its dependencies.
- **Docker container:** A running instance of a Docker image.

## How do you create and run a Docker container?

```
docker run -d --name my-container -p 8080:80 my-image
```

4.

- `-d`: Run in detached mode (background).
- `--name`: Container name.
- `-p`: Port mapping (host:container).

## What is Dockerfile?

A Dockerfile is a script containing a series of commands (instructions) to build a Docker image.  
Example:

```
dockerfile
CopyEdit
FROM openjdk:17
COPY . /app
WORKDIR /app
RUN javac App.java
CMD ["java", "App"]
```

5.

---

## Intermediate Docker Questions:

### 6. What is the difference between CMD and ENTRYPOINT in Docker?

- **CMD:** Provides default commands/arguments for a container. Can be overridden.
- **ENTRYPOINT:** Sets a fixed command that **cannot** be overridden.  
Example:

```
CMD ["npm", "start"]
```

```
ENTRYPOINT ["java", "-jar", "app.jar"]
```

7.

### How do you check running Docker containers?

```
docker ps
```

8. Use `docker ps -a` to view all containers (running and stopped).

### How do you stop a running Docker container?

```
docker stop <container_id>
```

9.

### What is the purpose of the EXPOSE command in Dockerfile?

It informs Docker that the container listens on a specified port at runtime. It doesn't publish the port — use `-p` during `docker run`.

Example:

```
EXPOSE 8080
```

10.

### What is Docker Compose?

Docker Compose is a tool to define and run multi-container Docker applications using a `docker-compose.yml` file.

Example:

```
yaml
CopyEdit
version: '3'
services:
  app:
    image: my-app
    ports:
      - "8080:8080"
```

Run using:

```
bash
CopyEdit
docker-compose up
```

11.

---

## Advanced Docker Questions:

### 11. What is the difference between Docker Volumes and Bind Mounts?

- **Volumes:** Managed by Docker, stored in Docker's storage location.
- **Bind Mounts:** Maps a host file/directory to a container file/directory.  
Example of volumes:

```
bash
CopyEdit
docker run -v my-volume:/app/data my-image
```

12.

### 13. How do you copy files between host and Docker container?

**Host to container:**

```
bash
CopyEdit
docker cp file.txt my-container:/app
```

○

### Container to host:

```
bash
CopyEdit
docker cp my-container:/app/file.txt .
```

○

### What is a multi-stage build in Docker? Why is it useful?

Multi-stage builds allow you to create lean production images by separating build and runtime stages.

Example:

```
dockerfile
CopyEdit
FROM maven:3.8.5 AS build
WORKDIR /app
COPY . .
RUN mvn package

FROM openjdk:17
COPY --from=build /app/target/app.jar app.jar
CMD ["java", "-jar", "app.jar"]
```

14. **Why use it?** To reduce image size by excluding unnecessary build dependencies.

15. **How do you clean up unused Docker images, containers, and volumes?**

Remove all stopped containers:

```
bash
CopyEdit
docker container prune
```

○

Remove all unused images:

```
bash
CopyEdit
docker image prune
```

○

Remove everything (use cautiously!):

bash  
CopyEdit  
`docker system prune -a`

○

## 16. What is the difference between a Docker network and a bridge network?

- **Docker network:** A way to connect multiple containers to communicate securely.
- **Bridge network:** The default network that allows containers on the same host to communicate.  
Create a custom bridge network:

bash  
CopyEdit  
`docker network create my-bridge`

17.

---

## ✓ Scenario-Based and Conceptual Questions:

### 16. How do you debug a Docker container?

To get logs:

bash  
CopyEdit  
`docker logs <container_id>`

○

To access the running container:

bash  
CopyEdit  
`docker exec -it <container_id> /bin/bash`

○

Check container health (if health checks are defined):

bash

CopyEdit

```
docker inspect --format='{{json .State.Health}}' <container_id>
```

○

### What are Docker secrets?

Docker secrets are a secure way to manage sensitive data (like API keys, passwords) for Docker services in **Swarm mode**.

Example:

bash

CopyEdit

```
docker secret create db_password password.txt
```

17.

18. **How would you deploy a Java Spring Boot app using Docker?**

Steps:

Create a **Dockerfile**:

dockerfile

CopyEdit

```
FROM openjdk:17
```

```
COPY target/app.jar app.jar
```

```
CMD ["java", "-jar", "app.jar"]
```

○

Build the Docker image:

bash

CopyEdit

```
docker build -t spring-boot-app .
```

○

Run the container:

bash

CopyEdit

```
docker run -p 8080:8080 spring-boot-app
```

○

### How do you link Docker containers?

Use Docker networks:

bash

CopyEdit

```
docker network create my-network
```

```
docker run --network=my-network --name db mysql
```

```
docker run --network=my-network --name app spring-boot-app
```

19. Containers can now communicate via their names (e.g., `db:3306`).

### 20. How do you implement CI/CD with Docker?

**Build Docker image** in CI pipeline (e.g., GitHub Actions, Jenkins):

yaml

CopyEdit

steps:

```
- name: Build Docker image
  run: docker build -t my-app:latest .
```

```
- name: Push to Docker Hub
  run: docker push my-app:latest
```

○

○ **Deploy** using Docker Compose or Kubernetes in the CD pipeline.