

DOCKER INTERVIEW QUESTIONS:

1. What is the usage of Docker?

Docker is a containerization platform that allows to package an application with all its dependencies into one single entity as single container which can be easily deployed and run on any machine that supports docker.

Docker lets you package your applications along with all the necessary code, libraries, and dependencies into what's called a "container." This containerization process ensures that your application runs consistently across different environments, from your laptop to production servers.

2. What are the Features of Docker?

- **Containerization:** Docker enables you to package your applications and dependencies into lightweight, portable containers that can run consistently across different environments.
- **Isolation:** Each Docker container runs in its own isolated environment, ensuring that applications and their dependencies are encapsulated and don't interfere with other containers or the host system.
- **Portability:** Docker containers are highly portable, allowing you to easily move them between different environments, from development to production, without worrying about compatibility issues.
- **Efficiency:** Docker optimizes resource usage by sharing the host operating system's kernel among containers, resulting in faster startup times, reduced overhead, and improved resource utilization.
- **Scalability:** Docker makes it easy to scale your applications horizontally by running multiple instances of containers across multiple hosts, enabling you to handle increased traffic or workload seamlessly.

3. Can You tell What is the Functionality of a Hypervisor?

A hypervisor is a virtualization software that helps in running multiple operating systems (Guest OS) on a single physical host system by providing an isolation between the virtual machines (VMs) and manages their resources.

4. Difference between Docker and Virtualization?

Docker:

- Docker is a containerization platform that allows you to package and run applications along with their dependencies in isolated containers.
- Containers share the host operating system's kernel, making them lightweight and efficient.
- Docker containers are portable and can run consistently across different environments.

Virtualization:

- Virtualization involves creating virtual machines (VMs) that mimic physical hardware, allowing you to run multiple operating systems and applications on a single physical machine.
- Each VM has its own virtualized hardware, including CPU, memory, storage, and network interfaces.

5. Docker images?

Docker images are the blueprints or templates used to create Docker containers. They contain everything needed to run a containerized application, including the application code, runtime, libraries, dependencies, and configuration files.

Base Images: Docker images are typically based on a base image, which provides the foundation for the container. Base images are often minimal operating system distributions, such as Alpine Linux or Ubuntu, or specialized images for specific purposes, such as web servers or databases.

6. Docker Containers?

Docker Containers: Docker containers are standardized , encapsulated environments that run applications/instances of Docker images.

7. Docker Engine?

Docker Engine: Docker engine is the runtime that executes containers.

8. What is Docker Hub?

Docker Hub is container registry that serves as a centralized repository for Docker images. It built for developers and open source contributors to find , use , share and download container images. Docker Hub can be used either host public repos that can be used for free, or docker private repos for teams and enterprises.

9. What are the steps to create a container?

- Write a Dockerfile:
Image:docker build -t image_name .
- Build the Docker
Image:docker build -t image_name .
- Run the Container
Docker run -d --name container_name image_name

10. How we can access the container?

Access the Container: You can interact with the container using the docker exec command to run commands inside the container. For example:

```
docker exec -it container_name bash
```

(or)

```
Docker start container
```

```
Docker attach container
```

11. How to Start, Stop, and Kill a Container?

In Docker to start, stop and kill a container we using start, stop and kill options on association with the docker command , the usage is given below.

- **docker start < container_name >**
- **docker stop < container_name >**
- **docker kill < container_name >**

12. What is Dockerfile?

A Dockerfile is a text file that contains a set of instructions for building a Docker image. It defines the steps needed to create a Docker image, including specifying the base image, installing dependencies, copying files, setting environment variables, and running commands.

```
CMD > Dockerfile
1 FROM almalinux:8
2 RUN yum install nginx -y
3 RUN echo "welcome to Dockerfile" > /usr/share/nginx/html/hello.html
4 CMD ["nginx", "-g", "daemon off;"]
```

13. what are the instructions in docker file?

- FROM: Sets the base image for subsequent instructions.
- LABEL: Adds metadata to the image as key-value pairs.
- MAINTAINER: Deprecated. Use LABEL instead.
- RUN: Executes commands in the container during build.
- CMD: Specifies the default command to run when the container starts.
- EXPOSE: Specifies the ports on which a container listens for connections.
- ENV: Sets environment variables in the container.
- ADD: Copies files, directories, or remote URLs into the image.
- COPY: Copies files or directories into the image.
- ENTRYPOINT: Specifies the executable to run when the container starts.
- VOLUME: Creates a mount point for externally-mounted volumes.
- USER: Sets the user or UID for the container.
- WORKDIR: Sets the working directory for subsequent instructions.
- ARG: Defines build-time variables.

14. Can You Tell the Difference Between CMD and ENTRYPOINT?

CMD vs ENTRYPOINT

CMD is used for setting default commands and arguments that will be executed at the start of runtime of the containers. It is often overridden by providing command-line arguments during container startup.

ENTRYPOINT configures a container to run as an executable, defining the command that has to be executed when the container starts. It is stricter than CMD and is often used when the container should have to behave like an executable.

15. ADD Vs COPY?

COPY:

The COPY instruction copies files or directories from the host machine into the Docker image. It has a straightforward behavior and is typically used to copy local files and directories into the image.

ADD:

The ADD instruction has additional features compared to COPY.

In addition to copying files and directories, it also supports copying files from URLs and unpacking archives (e.g., .tar, .tar.gz, .tar.bz2, .tar.xz, .zip). It has a more complex behavior, which may lead to unexpected results in certain cases.

Because of its additional features, it's generally recommended to use COPY instead of ADD unless you specifically need the additional functionality provided by ADD.

16. Where are Docker Volumes Stored in Docker?

Docker volumes provide a way to persist data generated by and used by Docker containers. They offer several advantages over bind mounts or copying data into containers:

- Data Persistence: Volumes persist data even if the container is removed or stopped. This makes volumes suitable for storing important data, such as databases or configuration files, that you don't want to lose when a container is deleted.
- Sharing Data Between Containers: Volumes can be shared between multiple containers, allowing data to be easily shared and accessed by different parts of an application or different services.

17. Name the Essential Docker Commands and What They Do?

The essential Docker Commands are listed here:

- **docker run:** Docker run command is used to run the docker image as a docker container.
- **docker ps:** Docker ps command will list all the running container in the docker.
- **docker exec:** It helps in execute the commands in a running container.
- **docker stop:** It will stop a container which is running in the docker.

18. How Do You Share Data Between Containers in Docker?

In Docker, you can share data between the containers on using volumes or by utilizing the `--volumes-from` option. Volumes will provide a persistent and the shared storage mechanism, allowing the data to be accessed and modified by multiple containers.

19. What is Docker Swarm, and how does it work?

Docker Swarm is a native clustering and orchestration tool provided by Docker. It allows you to create and manage a cluster of Docker hosts, turning them into a single virtual Docker host. (BUT now a days it is not using just know for the interview purpose).

20. ARG usage in dockerfile?

The ARG instruction in a Dockerfile is used to define build-time variables that can be passed to the Docker build process.

Pls refer all the commands from classes