

## Important Docker Interview Questions in 2023

### Description

## What is containerization

Container is an environment that is isolated from the rest of your operating system. Containerizing an application or software means packaging the code, configuration and its dependencies together required to run the application. By doing this you don't have to depend on the server configuration which may be incompatible with the application. By containerizing the application, we can run the container in any operating system which has container runtime installed in it (Example- Docker).

To understand the concept in simple terms, let's look at the below image. We have a blue laptop which represents a Server which has its own Host operating system. Then we have an orange laptop inside which represents a container. In this container I can install anything like MongoDB, Postgres, JDK and it does not affect the laptop which has the Host operating system.



More information on container can be [found here](#).

## What is a docker image

Images are basically read-only templates with instructions to create docker containers. We can get the images from the Docker hub and create containers out of it. We can also create our own images by creating dockerfile. Inside the dockerfile we will write all the instructions which will create a container with required artifacts.

You can relate the image as the recipe for any food/dish. And the end result as the container containing the food. More information on docker image can be [found here](#)

## What is a docker container

Docker containers are the lightweight alternatives of the virtual machine. It allows developers to package the application with all its dependencies and libraries, and ship it as a single package. The advantage of using a docker container is that you don't need to allocate any RAM and disk space for the applications. It automatically generates storage and space according to the application requirement.

Advantages of a container is that it is not specific to any operating system. It can be used across OS. It can be ported on to a Linux machine or a windows machine or even in cloud machine.

More information on container can be [found here](#).

## What is a docker engine

Docker daemon or docker engine is a process responsible to create docker images, container etc.

## What are docker components

**Docker client:** Docker client performs docker build, pull and run operations to open connection with the docker host.

**Docker host:** The Docker host contains the docker daemons, containers and its associated images. The Docker daemons establishes a connection with the registry.

**Registry:** Registry is where all the images are stored. Public registry and Private registry are two types of registries. Docker hub and Docker cloud are are two public registries.

## What is docker registry and what are types of docker registry

Docker registry is a public registry where the images are stored.

There are two types of registries, public registry and private registry. Docker Hub is the public registry where all the images are stored. We can also create our own private registries. With the help of docker commands the images can be pulled or pushed to these registries.

## What is a Dockerfile

Dockerfile is a file which contains all the commands or instructions which are required for building an image. One way to run the docker commands are through the command line. The same list of instructions can be written in Dockerfile and we can build the images to create the docker containers.

Docker can build the images automatically by reading the instructions from the Dockerfile.

## How to run Dockerfile

After creating the Dockerfile we need to build it in order to create an image.

```
docker build <docker file path>
```

## What is a docker volume

Docker volume are the directories present in the container. We can synchronize the data present in the container directory with any of the host system directory. The data synchronization is required because when we delete the container, the data will also get deleted. Hence data security is take care of.

While running the docker command in the command line we use **-v** volume parameter. Syntax of docker volume command is as below.

```
docker run -v host_path:docker_path <container_name>
```

In the below example, /data/app directory is mounted to /usr/src/app host directory.

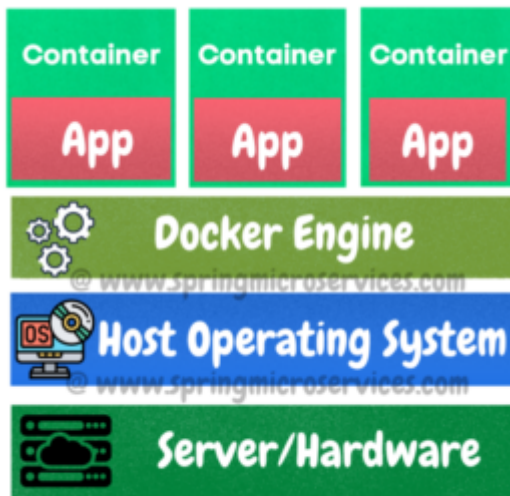
```
docker run -v /data/app:usr/src/app myapplication
```

## Where are docker volumes stored in docker

Docker volumes are stored in the Docker host filesystem at below path. Volumes are completely managed and created by Docker.

```
/var/lib/docker/volumes
```

## What is docker architecture



**Docker Engine** – Docker engine is a docker process which runs in the background. Docker engine is required for docker related activities like building the images, creating containers, creating volumes etc.

**Docker REST API** – Docker REST APIs are the interface by which we can talk to the Docker daemon.

**Docker CLI** – Docker CLI uses Docker REST API to interact with Docker daemon

## What is docker compose

Docker compose is a tool used to make running multiple docker containers with all the configuration, much easier as compared to executing docker run commands every time. By using Docker Compose, we can define the services in a docker-compose.yml file, as well as spin them up and tear them down with one single command. We usually create containers by running docker commands on the command line. But always running the docker commands is a difficult task. It becomes even more tedious if there are multiple containers to be created. Hence Docker compose is used to automate the task of container creation, establish the connectivity between them and various other container related task.

## How to install docker compose

The easiest and recommended way to get Docker Compose is to install Docker Desktop. Docker Desktop includes Docker Compose along with Docker Engine and Docker CLI which are Compose prerequisites. If you have already installed Docker Desktop, you can check which version of Compose you have by selecting About Docker Desktop from the Docker menu

## What is docker hub

Docker hub is the public cloud based image registry provided by Docker. Docker hub provides features

of storing and sharing the images publicly.

## What are the basic system requirements for the docker to run

Docker can be run on both Windows as well as Linux platforms.

- For Windows platform, docker needs at least Windows 10 64bit with 2GB RAM space. For the lower versions, docker can be installed by taking help of the toolbox. Docker can be downloaded from <https://docs.docker.com/docker-for-windows/> website.
- For Linux platforms, Docker can run on various Linux flavors such as Ubuntu >=12.04, Fedora >=19, RHEL >=6.5, CentOS >=6 etc.

## What is docker swarm

Docker swarm is basically a container orchestration tool. Container orchestration is basically managing cluster of containers effectively. Apart from Docker swarm, other container orchestration tools available are Kubernetes and Apache Mesos. Docker swarm is used for managing thousands or even more containers running on different machines.

## What is a docker namespace

Docker namespace is one of the core concepts in containerization as they introduce a layer of isolation among the container. In docker, the namespaces ensure the containers are portable and they don't affect the host. Namespace types support by docker are Network, PID, Mount, IPC, User.

## What is the lifecycle of the docker container

Lifecycle of docker container contains all the stages right from start of creating the container up till the end of container. Below are each stages of docker lifecycle.

- Create container
- Run container
- Pause container
- Unpause container
- Start container
- Stop container
- Restart container
- Kill container
- Destroy container

## How do you get the number of containers running, paused and stopped?

You can get the number of containers running, paused, stopped, the number of images and a lot more

by using the below command.

```
docker info
```

## How to login into docker repository

We can use below command to login into docker repository

```
docker login
```

## How to create a docker container from a docker image

Below command pulls the image from the docker repository if not present locally, and the created a container

```
docker run -it -d <image_name>
```

## How to list all the running containers

Below command list all the running containers.

```
docker ps
```

## How to start, stop and kill a container

Below command is used to start a docker container

```
docker start <container_id>
```

Below command is used to stop a docker container

```
docker stop <container_id>
```

Below command is used to kill a docker container.

```
docker kill <container_id>
```

## How to delete a stopped container

Below command is used to delete a stopped container

```
docker rm <container_id>
```

## How to delete an image from the local storage

Below command is used to delete an image from the local storage

```
docker rmi <image_id>
```

## How is the use of docker system prune command.

```
docker system prune
```

docker system prune command is used to remove all the stopped containers, all the networks that are not used and all the dangling images and build caches.

## How to monitor docker in production

We can monitor docker in production by using docker commands like docker stats and docker events. Docker stats provides information like CPU and memory usage of the container. Docker events provides information about all the activities going on in the docker daemon.

## What is the difference between docker and docker compose

Docker CLI is used for managing individual containers on a docker engine. Docker command line is used for accessing the docker daemon api. A Dockerfile is a document which contains all the commands or instructions a user would call on the command line to assemble an image.

Docker-compose is used for managing multi-container applications. Whatever commands we run on the docker CLI can be moved into the docker-compose.yml for easier reuse.

### Category

1. Docker
2. Interview

**Date Created**

February 16, 2023

**Author**

kk-ravi144gmail-com

default watermark