What is a Docker?

A Docker is defined as the platform for containerizing the applications to isolate it from each other in order to ensure high availability and more efficiency irrespective of the environments such as Development, Testing or Production. All the application related dependencies such as libraries, jar files, server related configurations, infrastructure-related elements will be packaged and formed as container called containerized application which does not need any dependency and works independently. It ensures the application to be run irrespective of the external factors. Containers in Docker have support from Docker Engine and Host Operating System to support all the operational or infrastructural related dependencies.

What is Hypervisor?

A hypervisor is a software that makes virtualization possible. It is also called Virtual Machine Monitor. It divides the host system and allocates the resources to each divided virtual environment. You can basically have multiple OS on a single host system.

What is virtualization?

Virtualization is the process of creating a software-based, virtual version of something(compute storage, servers, application, etc.). These virtual versions or environments are created from a single physical hardware system. Virtualization lets you split one system into many different sections which act like separate, distinct individual systems. A software called Hypervisor makes this kind of splitting possible. The virtual environment created by the hypervisor is called Virtual Machine.

What is containerization?

Let me explain this is with an example. Usually, in the software development process, code developed on one machine might not work perfectly fine on any other machine because of the dependencies. This problem was solved by the containerization concept. So basically, an application that is being developed and deployed is bundled and wrapped together with all its configuration files and dependencies. This bundle is called a container. Now when you wish to run the application on another system, the container is deployed which will give a bug-free environment as all the dependencies and libraries are wrapped together. Most famous containerization environments are Docker and Kubernetes.

What are the components of Docker Architecture and explain?

This is the common Docker Interview Questions asked in an interview. The Docker works on client-server architecture. The Docker client establishes communication with the Docker Daemon. The Docker client and Daemon can run on the same system. A Docker client can also be connected to a remote Docker Daemon. The different types of Docker components in a Docker architecture are–

1. **Docker Client:** This performs Docker build pull and run operations to establish communication with the Docker Host. The Docker command uses Docker API to call the queries to be run.

2. **Docker Host**: This component contains Docker Daemon, Containers and its images. The images will be the kind of metadata for the applications which are containerized in the containers. The Docker Daemon establishes a connection with Registry.

3. **Registry:** This component will be storing the Docker images. The public registries are Docker Hub and Docker Cloud which can be s used by anyone.

What is Docker Container?

A Docker Container is a form of encapsulation to the application which holds all the dependencies which share the kernel with other containers in the duration of running the isolated processes on the host operating system. A Docker container can be created by creating a Docker image. These Docker images can be run after that using Docker commands. Docker containers are the instances of the Docker images at the runtime. Docker images can be stored in any public hosts or private hosts like Docker hub. Docker Image is a set of files which can be run in an isolated process.

What are Docker Images?

When you mention Docker images, your very next question will be “what are Docker images”. Docker image is the source of Docker container. In other words, Docker images are used to create containers. When a user runs a Docker image, an instance of a container is created. These docker images can be deployed to any Docker environment.

What is a Docker Registry?

A Docker Registry is a place where all the Docker Images will be stored and Docker Cloud and Docker Hub are the public registries where these images can be hosted upon. The Docker hub is the default storage for the Docker Images. An own registry can also be set up as per the requirement. Docker Data Center (DDC) can also be used which includes DTR (Docker Trusted Registry). Docker store will provide the feature of buying and selling the Docker images.

What is the lifecycle of Docker Container?

This is the most popular Docker Interview Questions asked in an interview. The life cycle of the Docker container is as below:

1. Create a container.
2. Run the Docker container.
3. Pause the Container.
4. Unpause the Container.
5. Start the Container.
6. Stop the Container.
7. Restart the Container.
8. Kill the Container.
9. Destroy the Container.

Difference between virtualization and containerization

Once you've explained containerization and virtualization, the next expected question would be differences. The question could either be differences between virtualization and containerization or differences between virtual machines and containers. Either way, this is how you respond.

Containers provide an isolated environment for running the application. The entire user space is explicitly dedicated to the application. Any changes made inside the container is never reflected on the host or even other containers running on the same host. Containers are an abstraction of the application layer. Each container is a different application.

Whereas in Virtualization, hypervisors provide an entire virtual machine to the guest(including Kernal). Virtual machines are an abstraction of the hardware layer. Each VM is a physical machine

What is Docker Hub?

Docker images create docker containers. There has to be a registry where these docker images live. This registry is Docker Hub. Users can pick up images from Docker Hub and use them to create customized images and containers. Currently, the Docker Hub is the world's largest public repository of image containers.

What is a Dockerfile?

Let's start by giving a small explanation of Dockerfile and proceed by giving examples and commands to support your arguments.

Docker can build images automatically by reading the instructions from a file called Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. Using docker build, users can create an automated build that executes several command-line instructions in succession.

The interviewer does not just expect definitions, hence explain how to use a Dockerfile which comes with experience. Have a look at this tutorial to understand how Dockerfile works.

Tell us something about Docker Compose.

Docker Compose is a YAML file which contains details about the services, networks, and volumes for setting up the Docker application. So, you can use Docker Compose to create separate containers, host them and get them to communicate with each other. Each container will expose a port for communicating with other containers.

How to check for Docker Client and Docker Server version?

The following command gives you information about Docker Client and Server versions:

```
$ docker version
```

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

You can use the following command to get detailed information about the docker installed on your system.

```
$ docker info
```

If you wish to use a base image and make modifications or personalize it, how do you do that?

You pull an image from docker hub onto your local system

It's one simple command to pull an image from docker hub:

```
$ docker pull <image_name>
```

How do you create a docker container from an image?

Pull an image from docker repository with the above command and run it to create a container. Use the following command:

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

Most probably the next question would be, what does the '-d' flag mean in the command?

-d means the container needs to start in the detached mode. Explain a little about the detach mode. Have a look at this blog to get a better understanding of different docker commands.

How do you list all the running containers?

The following command lists down all the running containers:

```
$ docker ps
```

How to start, stop and kill a container?

The following command is used to start a docker container:

```
$ docker start <container_id>
```

and the following for stopping a running container:

```
$ docker stop <container_id>
```

kill a container with the following command:

```
$ docker kill <container_id>
```

Once you've worked with an image, how do you push it to docker hub?

$ docker push <username/image name>

How to delete a stopped container?

Use the following command to delete a stopped container:

```
$ docker rm <container id>
```

How to delete an image from the local storage system?

The following command lets you delete an image from the local system:

```
$ docker rmi <image-id>
```

How to build a Dockerfile?

Once you've written a Dockerfile, you need to build it to create an image with those specifications. Use the following command to build a Dockerfile:

```
$ docker build <path to docker file>
```

The next question would be when do you use ".dockerfile_name" and when to use the entire path?

Use ".dockerfile_name" when the dockerfile exits in the same file directory and you use the entire path if it lives somewhere else.