1. **What is docker?**

Docker (dock worker) is used to create containers which is standard way of packaging any application.

(Or)

Docker is a containerized technology it is used to build, run and test.

1. **What is docker image?**

If the application can be developed in any sever by the standard way of packaging.

(or)

A docker image is a read only template that contains a set of instruction for creating a container that can run on the docker platform.

(or)

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.

1. ****What is Hypervisor?****

A Hypervisor is a software to create virtual machines. 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. There are two types of Hypervisors:

* Type 1: It’s also called Native Hypervisor or Bare metal Hypervisor. It runs directly on the underlying host system. It has direct access to your host’s system hardware and hence does not require a base server operating system.
* Type 2: This kind of hypervisor makes use of the underlying host operating system. It’s also called Hosted Hypervisor.

### ****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](https://hub.docker.com/" \t "https://www.edureka.co/blog/interview-questions/docker-interview-questions/_blank) is the world’s largest public repository of image containers.

### ****What is a Dockerfile?****

Docker file is a simple text file that consist of instructions to build a docker image.

(or)

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.

### ****What is Namespace?****

Isolation's on the Linux machines are created using a Linux kernel feature called Namespaces.

PID namespaces (process namespaces)

It creates the isolated process tree inside the container.

NET namespaces (NETWORK namespaces)

It creates the isolated network for each container form network interfaces.

MOUNT namespaces

It allows each container to have their own file system view which starts from root.

USER namespaces

It allows to create whole new set of user and groups for the container.

### ****What is the lifecycle of a Docker Container?****

Docker containers have the following lifecycle:

* Create a container
* Run the container
* Pause the container(optional)
* Un-pause the container(optional)
* Start the container
* Stop the container
* Restart the container
* Kill the container
* Destroy the container

### ****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

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

### ****How to login into docker repository?****

You can use the following command to login into hub.docker.com:

$ docker login

### ****How do you list all the running containers?****

The following command lists down all the running containers:

$ docker ps

### ****How do you access a running container?****

The following command lets us access a running container:

$ docker exec -it <container id> bash

The exec command lets you get inside a container and work with it.

### When should I use Dockerfile?

One should use a Dockerfile when there’s a need to distribute/collaborate on the app’s operating system with a team. Use Dockerfile as the version control system for the entire app’s OS. Or use Dockerfile to run the code to the laptop in the same environment as the server you are working on.

### Is Dockerfile a text file?

Dockerfile is a text document containing all the commands the user requires to call on the command line to assemble an image. With the help of a Dockerfile, users can create an automated build that executes several command-line instructions in succession.

### How do I create a simple Dockerfile?

To create a Dockerfile, set up Docker and Docker Hub. Create the original Docker container and then create a file on it. Make changes to the container, and finally, create a new image.

### What is Dockerfile language?

Go language is used to write Docker. A Dockerfile is a text file that contains collections of instructions and commands that will be automatically executed in sequence in the docker environment for building a new docker image.

### What is Docker compose vs. Dockerfile?

The key difference between Docker compose, and Docker is that the Docker contents describe how to create and build a Docker image, while Docker compose is a command that runs Docker containers based on settings described in a docker-composed.yaml file.

### What is a .dockerignore file?

A .dockerignore file allows you to specify a list of files or directories that Docker is to ignore during the build process. It is similar to a .gitignore file, which is used when you build Git repositories. You can specify the list of files and directories inside the .dockerignore file.

### How do I commit a docker container?

First, you need to pull a docker image. Then deploy the container, modify it and commit the changes to the image. When you commit to changes, you essentially create a new image with an additional layer that modifies the base image layer.

### What is docker context?

With a docker context, importing and exporting context on different machines with the Docker installed gets easier. One can also use a docker context export command to export an existing context to a file, which can be imported to another machine with the docker client installed.

### What is the use of Docker in DevOps?

A Docker container image in DevOps is a lightweight, executable, and standalone package of software that includes everything needed to run an app, runtime, system tools, settings, system libraries, and code. Containers simplify the build/test/deploy pipelines in DevOps.

### Is Dockerfile COPY recursive?

COPY and ADD are the two commands that Docker provides for copying files from the host to the Docker image when building it. COPY command copies files recursively, given explicit source and destination directories or files.

### Is Dockerfile a Docker image?

A Docker image is built automatically by reading the instructions from a Dockerfile, a text file containing all commands needed to build a given image. A Dockerfile adheres to a specific set of instructions and format, producing a Docker image when you build it.

1. **What is SWARM?**

Multi-host networking is created as part of docker orchestration called as SWARM.

1. **What is overlay network?**

Docker has a network driver called as overlay network.

1. **What is continuous deployment?**

Continuous deployment is basically when terms rely on fully-automated pipeline. This practice fully eliminates any manual steps and automates the entire process. Threrfore, countinuous deployment ensures that code is countinuously being pushed into production.

1. **What is continuous delivery?**

Continuous delivery is an ongoing DevOps practice of building, testing, and delivering improvements to software code and user environments with the help of automated tools.

##What are different ways of building docker images?

**Build Docker Image Using Dockerfile**

Step 1: Create the required Files and folders.

Step 2: Create a sample HTML file & Config file.

Step 3: Choose a Base Image.

Step 3: Create the Dockerfile.

Step 4: Build your first Docker Image.

Step 5: Test the Docker Image.

##How is container different than VM?

A container is a software code package containing an application's code, its libraries, and other dependencies. Containerization makes your applications portable so that the same code can run on any device. A Virtual machine is a digital copy of a Physical machine.

##What is difference between ADD and COPY Instruction?

First, the ADD directive can accept a remote URL for its source argument. The COPY directive, on the other hand, can only accept local files. Note that using ADD to fetch remote files and copying is not typically ideal. This is because the file will increase the overall Docker image size.

##Can you explain the concept of Layers in Docker Image?

Layers are what make up an image. Each layer is a “diff” that contains the changes made to the image since the last one was added. When building an image, the platform creates a new layer for each instruction in the file. The container starts at the first instruction in the file and executes all instructions in order.

##What is the purpose of EXPOSE and VOLUME instruction in Dockerfile?

Basis for my last question: If anything, VOLUME and EXPOSE do serve as a sort of documentation. You can easily spot which aspects of the container's environment are relevant to the outside world. I suggest avoiding VOLUME , but the EXPOSE instruction is useful for documentation.

##What is your workflow for CI/CD with Docker Containers and where do you store images?

CI/CD Tool – An integration and test system that builds the docker image and runs a series of tests. It also pushes the built image to the Kubernetes cluster. Popular examples are Jenkins, Travis, or CircleCI. Kubernetes Cluster – Kubernetes deploy docker containers for the software build verified by the CI/CD tool.