# Docker

**------------------Docker------------------ Docker Image------------------Docker Container------------------**

**Docker** is essentially a toolkit that enables developers to manage(build, deploy, run, update, and stop) containers using simple commands. It is most used tool in microservices.

Docker provides the ability to package and run an application in a loosely isolated environment called a container.

The isolation and security allow you to run many containers simultaneously on a given host.

**What is difference between image and container?**

Java world : Class and Object.

Docker world : Image and Container.

Images can exist without containers, whereas a container needs to run an image to exist. Therefore, containers are dependent on images and use them to construct a run-time environment and run an application.

The two concepts exist as essential components (or rather phases) in the process of running a Docker container.

To make a docker image, you must write script in Dockerfile. ?? image or container

Containers are runnable instance of an image. You can create, start, stop, move, or delete a container using Docker API or CLI

Suppose a container is running an application at port 8080. It cannot access the host machine as each container is running in isolation. But we have to config to access the host.

Ansible provides seamless application configuration, while Docker provides a containerized environment for building and deploying applications. Therefore, you should consider Docker for code shipping and deployment and Ansible for application configuration.

How is memory management done in docker? How is it a lightweight when compared to Hypervisor?

A hypervisor allocates each VM resource such as CPU, memory, storage, and network to run the guest OS and applications--- How does the docker do it?

You can create containers without Docker, and you can also run and publish them.

What is in Dockerfile?

What does it do? how is it different from provisioning folder?

A document that contains commands to assemble an image. Docker reads those instructions and builds a docker image.

docker **pull** <image\_name>:<tag>

Or just

docker **run** <image\_name>:<tag> //this will pull and run the image.

docker **run** <image\_name>

docker run **-d** <image\_name> // detach mode. Run in background.

docker run **–name** <image\_name> //name the container

**Port binding with the container** :

docker run **-p**<machine\_port>:<container\_port> <name>:<tag>

docker **log** <container\_port>

docker **ps -a** // list running and stopped containers.

docker **start** <container\_ID> //to restart the container.

docker **exec** -it <container\_name> /bin/bash // to get the command line Terminal of container

docker **rm** <container\_name>

docker **rmi** <image\_name>

var/lib/docker //images , container location

**DockerFile**

# Use an OpenJDK base image with a specific version

FROM openjdk:11

# Set the working directory inside the container

WORKDIR /app

# Copy the JAR file of your Java application into the container

COPY target/your-java-app.jar /app/your-java-app.jar

# Specify the command to run when the container starts

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

*The purpose of specifying a base image is to provide a foundational environment on which you can add and configure your application and its dependencies. The base image contains an operating system (or a minimal runtime environment), along with any pre-installed software, libraries, and configurations needed to support your application.*

**Commands used in DockerFile**:

**ENTRYPOINT**:

*the entrypoint in Docker defines the initial command or process that runs when a container starts. It represents the primary purpose of the container and can be overridden or customized at runtime using the docker run command.*

*--entrypoint /bin/bash*

*This allows you to access the container's command line and interact with it as if it were a virtual machine or remote server.*

OpenShift : https://www.youtube.com/watch?v=kOjmmstohuI

Docker : <https://www.youtube.com/watch?v=HMAoJoSJCyk&list=PLRAV69dS1uWTJLvDP4Veld5F05rJAmOcp&index=3>