Docker 2 notes:

Base Image name:

FROM

It is used to specify base image for our application

FROM tomcat:9.0

FROM openJDK:17

FROM node:19.4

FROM mysql:8.5

FROM python:3.3

MAINTAINER

Maintainer decides the author of the Dockerfile

It is used to specify the author of the Dockerfile

MAINTAINER Abc [<abc@sldk.com>](mailto:<abc@sldk.com>)

First specify base image, followed by maintainer/author, run

RUN

At the time of Docker image creation, which instructions you want to execute

Run keyword is used to specify instructions to execute at the time of Docker image creation

Example: RUN ‘git clone<url>;’

RUN ‘mvn clean package’

If required, we can write multiple RUN instructions in a single Dockerfile and all those instructions will be processed in order

CMD

When does a Docker container get created?

When we try to execute the Docker image that time Docker container is created

CMD keyword is used to specify intructions to execute at the time of Docker container creation

At the time of image creation, we can write multiple RUN commands

CMD ‘java -jar app.jar’

CMD ‘app.py’

One Dockerfile can have multiple CMD instructions, however, Docker will process only last CMD instruction

RUN instructions will be executed at the time of Image creation, CMD instructions will be executed at the time of container creation

ENTRYPOINT

Keyword is used to specify instruction to execute at the time of docker container creation.

ENTRYPOINT is similar to CMD

ENTRYPOINT[“java”, “-jar”, “app.jar”]

ENTRYPOINT[“python”, “app.py”]

Note: CMD instructions we can override using command-line arguments however ENTRYPOINT instructions cannot be overridden

COPY

COPY any files from host machines to container machines

Docker container

Guest VM

Docker Engine

Linux

Host VM

COPY target/app.war /usr/app/tomcat/webapps/app.war

File must be available in host machine only then we can copy from host machine to guest VM

ADD

It is used to copy files from source to destination

ADD target/app.war /usr/app/tomcat/webapps/app.war

ADD <http-url> /usr/app/app.war

WORKDIR

COPY target/app.jar /usr/app/app.jar

WORKDIR /usr/app

It will go into /usr/app directory and execute further commands

CMD ‘java -jar app.jar’

EXPOSE

It is used to specify on which port# our application will run in our container

EXPOSE 8080 --> it is not to change port number it is only to provide information about port number to other team members