Rules of creating Dockerfile

FROM <image>:<tag>

FROM → The keyword that defines the base image.
<image> → The name of the Docker image/software
:<tag> (optional, defaults to latest) → The specific version of the image.

FROM <image>:tag

ex:

FROM python:3.9
FROM openjdk:17
FROM tomcat:9.0

MAINTAINER->it is used for specifies author name(deprecated)

ex:

MAINTAINER: Rahul rk <xyz@gmail.com>

ex:

LABEL maintainer="Rahul rk <xyz@gmail.com>"(Latest)

RUN command instruction in a Dockerfile is used to execute commands before build docker image

ex:

RUN 'git clone <repo-url>'

CMD command instruction in a Dockerfile is used to execute commands after build docker image

ex:

CMD "java -jar myapp.jar"

note: only last CMD command will executes even if you wright 5-times CMD command

ENTRYPOINT->Alternative command of CMD but the advantage of we cannot override this command (We give here executable command)

ex:

ENTRYPOINT [“java “, “-jar”. “myapp.jar”]

COPY instruction will copy the files from source to destination. i

ex:

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

Your paragraph text

Note if this path exist it will paste file here , if this path does not exist it will create path automatically

ADD

--> ADD instruction will copy the files from source to destination same as COPY, But in addition it can extract your compressed tar file
--> Here Source can be host machine
--> ADD cannot download from HTTP/S3 URLs—this is a misconception. (Be carefull in interviews)

Ex:
ADD target/app.jar /usr/app/
ADD <http-url> /usr/app/ (Wrong)

WORKDIR

--> The WORKDIR instruction sets the working directory (Like cd command)
--> If the directory does not exist, Docker will automatically create it

Ex: WORKDIR /path/to/directory

```
COPY target/your-app.jar   /usr/app/
WORKDIR /usr/app/
CMD "java -jar your-app.jar"
```

EXPOSE

-> EXPOSE command is used to specify application is running on which PORT number. Using this you cannot change the port number of application. You are just mentioning that our application is running on the port number 9090.

-> If our application is running on port number 8081 you are mentioning expose 9090 then it is wrong.

-> It is optional to use

Example : EXPOSE 9090

example

```
FROM openjdk:17

COPY target/demo-app.jar  /usr/app/

WORKDIR /usr/app/

EXPOSE 8080

ENTRYPOINT ["java", "-jar", "demo-app.jar"]
```