Instructions of Dockerfile

FROM: This instruction sets the base image.

ARG: This instruction defines a variable that can be set while building the image and will be available during image build. Passing the argument can be done using --build-arg <argument-value>. <https://docs.docker.com/engine/reference/builder/#arg> for official docs

LABEL <https://docs.docker.com/engine/reference/builder/#label>

ENV: With this instruction we can set environmental variable during image build as well when container is running. Environmental variables. <https://docs.docker.com/engine/reference/builder/#env>

WORKDIR: This instruction sets the workdir

RUN: This is most frequently used command, which does the activity of executing a command for some installation/configuration of your application.

COPY

ADD

USER: This instruction is used to specify the user so that all the steps/instructions after this will be executed as the user. USER has to be created before you use this instruction.

VOLUME

EXPOSE: This is used to inform Docker daemon about listening ports for containers.

CMD: This instruction defines the default process or argument when executing a container. The instructions in CMD can be written in shell form or exec form

* SHELL FORM: CMD java -jar spc.jar
* EXEC FORM: CMD ["java", "-jar", "spc.jar"]

ENTRYPOINT: This instruction will set which command container will run as executable. Whatever we write in CMD will be arguments to ENTRYPOINT

HEALTHCHECK

* For all instructions <https://docs.docker.com/engine/reference/builder/>
* Docker image is collection of read-only image layers

**Build a spring pet clinic application**

FROM alpine:3.18.2

LABEL author=shaikkhajaibrahim

ARG DOWNLOAD\_LOCATION=https://qtapps.s3.us-west-2.amazonaws.com/spring-petclinic-2.4.2.jar

ARG JAVA\_PACKAGE=openjdk17-jdk

RUN apk update && \

apk add ${JAVA\_PACKAGE}

RUN wget ${DOWNLOAD\_LOCATION}

Let’s build the image by changing args docker image build -t spc:v1.0 --build-arg "JAVA\_PACKAGE=openjdk11-jdk" --build-arg "DOWNLOAD\_LOCATION=https://referenceapplicationskhaja.s3.us-west-2.amazonaws.com/spring-petclinic-2.4.2.jar" .  


Let’s add an environmental variable in the docker image

FROM alpine:3.18.2

LABEL author=shaikkhajaibrahim

ARG DOWNLOAD\_LOCATION=https://qtapps.s3.us-west-2.amazonaws.com/spring-petclinic-2.4.2.jar

ENV JAVA\_TEST=hello

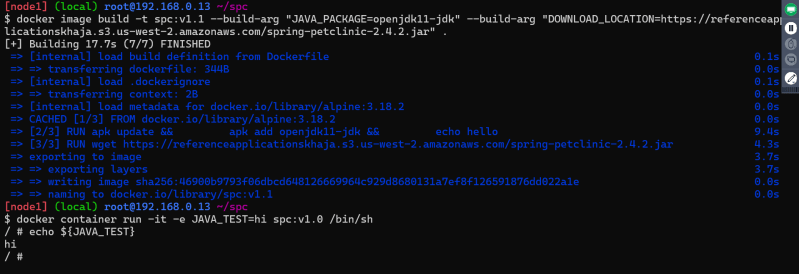
ARG JAVA\_PACKAGE=openjdk11-jdk

RUN apk update && \

apk add ${JAVA\_PACKAGE} && \

echo ${JAVA\_TEST}

RUN wget ${DOWNLOAD\_LOCATION}

Let’s create a container based on image. Lets change the environmental variables while running the container docker container run -it -e JAVA\_TEST=hi spc:v1.0 /bin/sh  


Let’s use CMD to start the application when container is started

FROM alpine:3.18.2

LABEL author=shaikkhajaibrahim

ARG DOWNLOAD\_LOCATION=https://referenceapplicationskhaja.s3.us-west-2.amazonaws.com/spring-petclinic-2.4.2.jar

ARG JAVA\_PACKAGE=openjdk11-jdk

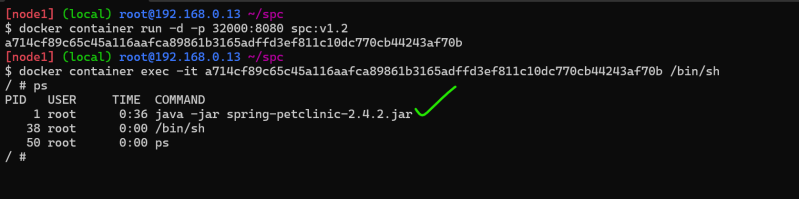
RUN apk update && \

apk add ${JAVA\_PACKAGE} && \

echo ${JAVA\_TEST}

RUN wget ${DOWNLOAD\_LOCATION}

CMD ["java", "-jar", "spring-petclinic-2.4.2.jar"]

run the container post building the image  


Let’s do the same thing in shell form

FROM alpine:3.18.2

LABEL author=shaikkhajaibrahim

ARG DOWNLOAD\_LOCATION=https://referenceapplicationskhaja.s3.us-west-2.amazonaws.com/spring-petclinic-2.4.2.jar

ARG JAVA\_PACKAGE=openjdk11-jdk

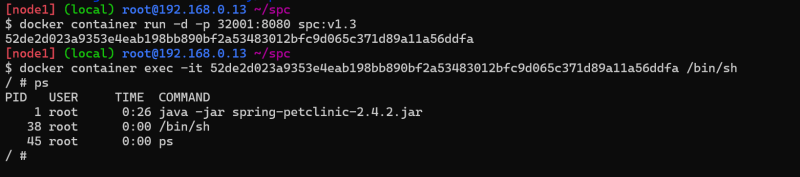
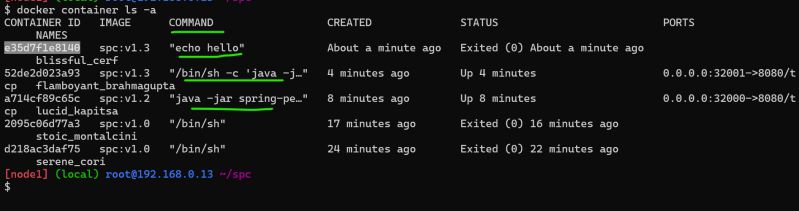
RUN apk update && \

apk add ${JAVA\_PACKAGE} && \

echo ${JAVA\_TEST}

RUN wget ${DOWNLOAD\_LOCATION}

CMD java -jar spring-petclinic-2.4.2.jar

  
\* Docker container will be in running state as long as main process is running  
\* Lets run the container docker container run -d spc:v1.3 echo hello  
  
\* echo hello has replace our CMD

**ENTRYPOINT**

FROM alpine:3.18.2

LABEL author=shaikkhajaibrahim

ARG DOWNLOAD\_LOCATION=https://referenceapplicationskhaja.s3.us-west-2.amazonaws.com/spring-petclinic-2.4.2.jar

ARG JAVA\_PACKAGE=openjdk11-jdk

RUN apk update && \

apk add ${JAVA\_PACKAGE} && \

echo ${JAVA\_TEST}

RUN wget ${DOWNLOAD\_LOCATION}

ENTRYPOINT ["java"]

CMD ["-jar", "spring-petclinic-2.4.2.jar"]

Let’s change this with EXPOSE

FROM alpine:3.18.2

LABEL author=shaikkhajaibrahim

ARG DOWNLOAD\_LOCATION=https://referenceapplicationskhaja.s3.us-west-2.amazonaws.com/spring-petclinic-2.4.2.jar

ARG JAVA\_PACKAGE=openjdk11-jdk

RUN apk update && \

apk add ${JAVA\_PACKAGE} && \

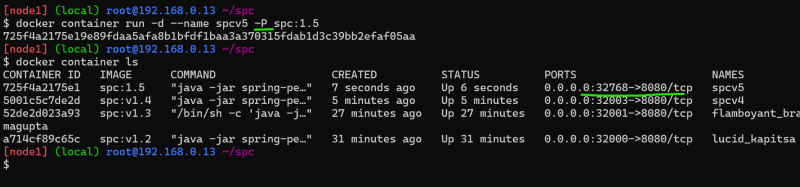
echo ${JAVA\_TEST}

RUN wget ${DOWNLOAD\_LOCATION}

EXPOSE 8080

ENTRYPOINT ["java"]

CMD ["-jar", "spring-petclinic-2.4.2.jar"]



**Security Concern**

* Never run your applications inside containers as root users, Create a USER and the run app inside container with that user

FROM alpine:3.18.2

LABEL author=shaikkhajaibrahim

ARG DOWNLOAD\_LOCATION=https://referenceapplicationskhaja.s3.us-west-2.amazonaws.com/spring-petclinic-2.4.2.jar

ARG JAVA\_PACKAGE=openjdk11-jdk

ARG USER=spc

ARG HOME\_DIR=/spc

ARG USER\_SHELL=/bin/sh

RUN apk update && \

apk add ${JAVA\_PACKAGE} && \

echo ${JAVA\_TEST}

RUN adduser -h ${HOME\_DIR} -s ${USER\_SHELL} -D ${USER}

USER ${USER}

WORKDIR ${HOME\_DIR}

RUN wget ${DOWNLOAD\_LOCATION}

EXPOSE 8080

ENTRYPOINT ["java"]

CMD ["-jar", "spring-petclinic-2.4.2.jar"]

Now start the container and exec into shell  
