**Docker Instructions**

ADD: Can download the contents from context/local folders as well as URLs, Recently support for git repos is also added

COPY: Can download the contents from context/local folders

Docker build context: This is set of the files located in specific PATH or URL specified during docker image build

**Copy instruction:** Using copy to copy the local file

FROM alpine:3.18.2

LABEL author=shaikkhajaibrahim

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}

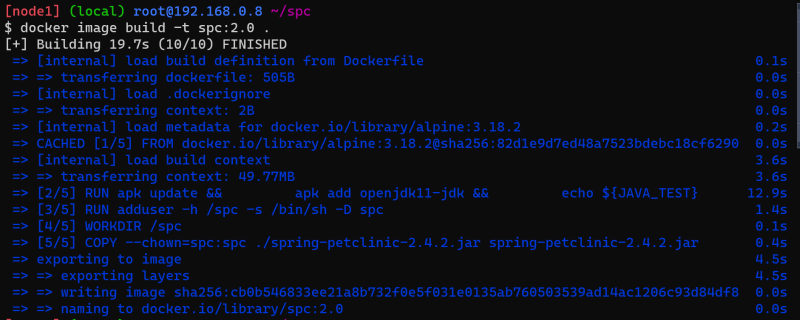
WORKDIR ${HOME\_DIR}

COPY --chown=${USER}:${USER} ./spring-petclinic-2.4.2.jar spring-petclinic-2.4.2.jar

EXPOSE 8080

ENTRYPOINT ["java"]

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

Now build the image  


If we use Add instruction i can directly use URL

FROM alpine:3.18.2

LABEL author=shaikkhajaibrahim

ARG JAVA\_PACKAGE=openjdk11-jdk

ARG USER=spc

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

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}

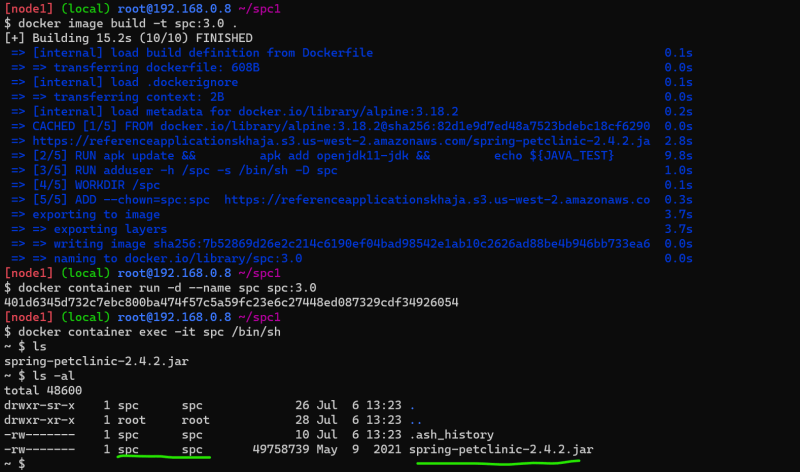
WORKDIR ${HOME\_DIR}

ADD --chown=${USER}:${USER} ${DOWNLOAD\_LOCATION} spring-petclinic-2.4.2.jar

EXPOSE 8080

ENTRYPOINT ["java"]

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

  
\* Let’s try to build the docker image of some python app <https://github.com/DevProjectsForDevOps/StudentCoursesRestAPI>   
\* Lets consider the following Dockerfile by using Add from git repo <https://docs.docker.com/engine/reference/builder/#adding-a-git-repository-add-git-ref-dir>

FROM python:3.7-alpine

LABEL author=KHAJA

LABEL blog=directdevops.blog

ARG HOME\_DIR='/studentcourses'

ADD https://github.com/DevProjectsForDevOps/StudentCoursesRestAPI.git $HOME\_DIR

ENV MYSQL\_USERNAME='directdevops'

ENV MYSQL\_PASSWORD='directdevops'

ENV MYSQL\_SERVER='localhost'

ENV MYSQL\_SERVER\_PORT='3306'

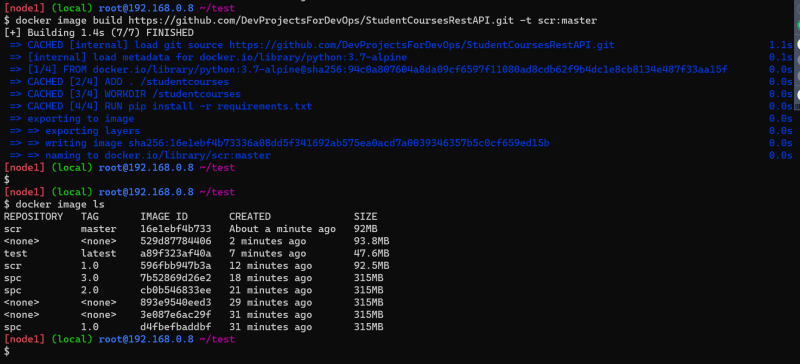
ENV MYSQL\_DATABASE='test'

EXPOSE 8080

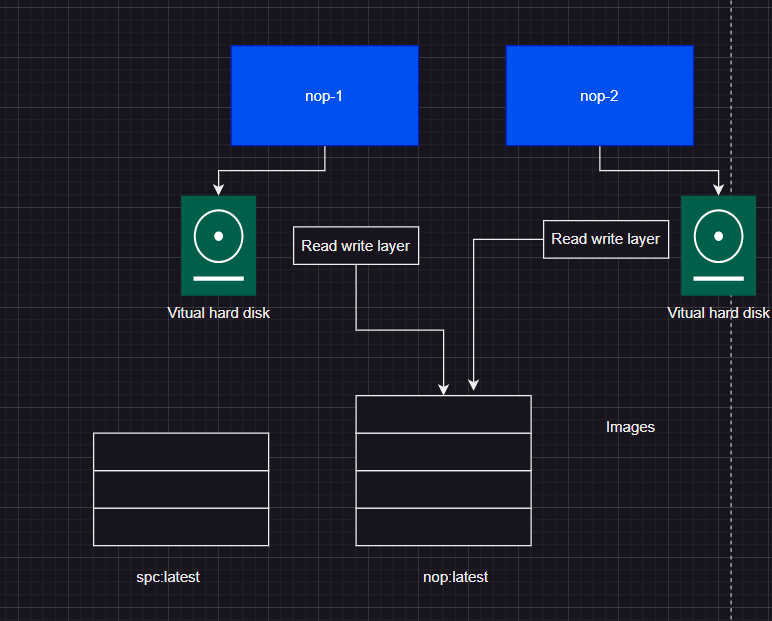
WORKDIR $HOME\_DIR

RUN pip install -r requirements.txt

ENTRYPOINT ["python", "app.py"]

* This was failing we will fix this
* We have built the images directly using docker image build <https://docs.docker.com/engine/reference/commandline/build/#git-repositories>   
  
* .DockerIgnore https://docs.docker.com/engine/reference/builder/#dockerignore-file

**Running Docker Containers**

* The storage in docker is based on union of layers, which is done by special storage drivers. Now docker majorly uses overlay as a storage driver.
* When the container is running the disk contents are union of all the read-only layers and one writable mounted as a single storage entity with the help of overlay  
  
* When the docker container is removed then we loose writable layer
* There are two kinds of applications
  + Stateless: These applications donot store state locally rather they rely on external systems such as databases, external mounts etc.
    - In modern applications all the microservices are stateless and they store state in some database
  + Stateful: These are applications which use local storage to store some data.
* To solve this problem with data loss, we have Docker volumes.
* Docker volumes ensure the data of the specific folder is persisted and has a life time beyond docker container