Lab Assignment 1 CMT

Max Shahrokni

TASK 2

Subtask 1

- Clone the repository

$ git clone https://github.com/yhl17cmt/Demo1WebApp.git

- Navigate into the cloned folder and create the Dockerfile with following content:

```
FROM maven:3.3-jdk-8 as builder
WORKDIR /code
COPY pom.xml /code
COPY src/ /code
RUN mvn clean install

FROM tomcat:9.0-alpine
COPY --from=builder /code/target/Demo1WebApp.war /usr/local/tomcat/webapps/
COPY --from=builder /code/main/webapp/index.jsp /usr/local/tomcat/webapps/ROOT/
EXPOSE 8080
CMD ["catalina.sh", "run"]
```

- You can now build the docker image from this multi-stage Dockerfile. This will in the first stage create the WAR file using maven, then in the second stage push the WAR file and index into the tomcat image that we will use to deploy the application. Following command will build the docker image mywebapp:

$ docker build -t mywebapp .

- Lets deploy the mywebapp image and connect the host port 8888 to container port 8080:

$ docker run -p 8888:8080 mywebapp

- The webapp is now accessible in a browser by URL:

http://HOST-IP:8888

Question 1

Why might you want to make a multi-stage build ?

Answer: In above case we have one Dockerfile with two FROM statements, it uses two separate images within one file to build the deployment image. The intermediary images are not required to run mywebapp, they may be removed which saves a lot of space.

Subtask 2

- Clone the repository:

$ git clone https://github.com/yhl17cmt/business-data.git

- - Navigate into the cloned folder and create the Dockerfile with following content:

```
FROM mariadb:latest
COPY Northwind_Database.sql docker-entrypoint-initdb.d
ENV MYSQL_ALLOW_EMPTY_PASSWORD=true
EXPOSE 3306
```

- You can now build the docker image from this Dockerfile. This will create the dockerimage maxdb with mariadb dbms, with the Northwind database:

$ docker build -t maxdb .

- Lets deploy the maxdb image and connect the host port 3307 to container port 3306:

$ docker run -p 3307:3306 maxdb

Subtask 3

- Make sure the machine your launching ansible on has ssh keys copied to the hosts running our containers. Create a project folder and navigate to it, create a textfile named "hosts" with following content:

```
[webserver]
remote_host_ip

[webserver:vars]
ansible_python_interpreter=/usr/bin/python3
```

- Next create a new textfile in the project folder named main.yml with following content:

```
- hosts: webserver
  remote_user: yourUserName
  tasks:
    - name: Running the web container
      docker_container:
        image: mywebapp:latest
        state: running

    - name: Running the DB container
      docker_container:
        image: maxdb:latest
        state: running
```