Lab 9: Selenium in Docker

Pre-requisites for executing the automation tests.

Go to following directory in your lab machine:

```
C:\Users\fenago\Desktop\advanced-selenium-java\Lab09
```

1. The project should be a maven project with a properly implemented pom.xml with all the project dependencies in it.

Step-1: To begin with we would want to have an image where we can ship our changes and containerize it. So, just type in **docker run ubuntu** Once you do that it will try searching your image locally. If it does not get an image it will download an image from the "*Docker registry*".

Once this is done you can check the images and the containers locally present in your host machine.

Step-2: Try running "**docker ps -a**" to get the list of all containers existing in your machine and running "**docker images**" will give you the list of all the images in the system. Check the image to refer

Step-3: Now what we need to do is configure a template which is basically called a "**Dockerfile**". The Docker client uses this template to interact with the daemon process. Create a file in your project root folder named as "**Dockerfile**".

In this `Dockerfile` we are going to write down few lines of code to create a container from the existing image which would preferably consists steps to install required dependencies and update the existing system.

Here is the code snippet that you would be needing to run this entire automation suite. In my project I have the `Dockerfile` inside the `Docker` directory.

```
# Use the latest image from UBUNTU installed in the machine
FROM ubuntu:latest

# Update ubuntu system
RUN apt-get update

# Install java version on ubuntu-selenium image
RUN apt-get install -y default-jdk
RUN apt-get install -y default-jre

# Install maven on ubuntu-selenium image
RUN apt-get install -y maven

# Install git on ubuntu-selenium image
RUN apt-get install -y git


WORKDIR /TestWebdriver
COPY TestWebdriver /TestWebdriver

# Note: You can also run following command if source is present in github public repo
# RUN git clone https://github.com/username/repo-name

# Run the maven command to execute all the tests
```

```
RUN mvn clean install test

ENTRYPOINT ["/bin/bash"]
```

So, this `dockerfile` would basically get the dependencies installed within the container which would include

1. `git`
2. `maven`
3. `Java`

After this it will take a clone of the required repository, navigate to the working directory and run the `mvn clean test` which would execute your required test automation within the container. Check it out in the image below. I have used OAUTH to get a clone of the repository. You might set the ssh key in the container as well.

So, in the image above what I did was docker build by using the command `docker build -t docker-selenium .` , -t represents tag name and . represents that the `Dockerfile` is in the current working directory. So, once you build has been execute you can even access the container. Just type in `docker run -it docker-selenium`

Note:

You can verify all the tools are present by running below commands:

```
java -version

mvn -version

git --version
```