Mule 4 Dockerization:

This document explains

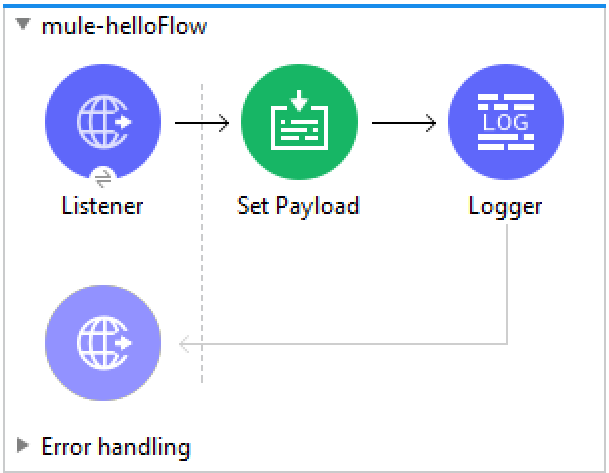
a) How to create a Mule 4.2.2 image using Windows docker desktop
b) How to run a sample Mule application on the same image.

Prerequisites:

1. Windows Docker Desktop is installed on your machine. https://hub.docker.com/search?q=&type=edition&offering=community&sort=updated_at&order=desc
2. Make sure you have switched to Linux based containers. Click on Docker Desktop icon and click on “Switch to Linux Containers”.
3. Anypoint Studio (7.4.2) is installed and you have a valid license (or trial) version of Mule Runtime “mule-ee-distribution-standalone-4.2.2.zip” placed in your system.

Let’s get our hands dirty:

1. Create a sample Mule application “mule-hello” and create a deployable archive out of it using Studio Export capability. We will not go in detail about this as this tutorial assumes that audience already has a brief knowledge of Mule based application development.



Configuration xml:

```
<?xml version="1.0" encoding="UTF-8"?>

<mule xmlns:http="http://www.mulesoft.org/schema/mule/http"
xmlns="http://www.mulesoft.org/schema/mule/core"
        xmlns:doc="http://www.mulesoft.org/schema/mule/documentation"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.mulesoft.org/schema/mule/core
http://www.mulesoft.org/schema/mule/core/current/mule.xsd
http://www.mulesoft.org/schema/mule/http
http://www.mulesoft.org/schema/mule/http/current/mule-http.xsd">
        <http:listener-config name="HTTP_Listener_config" doc:name="HTTP Listener config"
doc:id="db66e220-f8a2-4965-91e2-98c0bd9f2b7b" >
                <http:listener-connection host="0.0.0.0" port="8091" />
        </http:listener-config>
        <flow name="mule-helloFlow" doc:id="3c3cfc0c-27da-4eb5-8e7e-b4a796783308" >
                <http:listener doc:name="Listener" doc:id="15f952a9-2a9e-43d9-88a5-
56f49983ee8a" config-ref="HTTP_Listener_config" path="/hello"/>
```

```
                <set-payload value="Hello ! Welcome to world of Mule." doc:name="Set Payload"
        doc:id="7a5cd16f-fb44-42d2-b043-a39093ca2767" />
                <logger level="INFO" doc:name="Logger" doc:id="5f86155d-3bc3-4e54-b2fd-
        e1b49763f073" message="#[payload]"/>
            </flow>
            </mule>
```

This is a sample application which has a GET operation listening on 8091 port with action "/hello". This returns a pre-configured response in payload as "`Hello ! Welcome to world of Mule."`

2. Create a folder on your desktop to place the required items for this PoC. For example:

C:\Users\ankur\Desktop\Docker Demo

Place following items in this folder:

a) mule-ee-distribution-standalone-4.2.2.zip

b) mule-hello.jar

c) muleLicenseKey.lic (this is provided by your respective MuleSoft Customer Success Manager)

d) DockerFile

3. Now check if the Docker Desktop is up and running.

4. Populate DockerFile with following content:

DockerFile

```
FROM java:openjdk-8-jdk


MAINTAINER Ankur Parashar

#Add mule runtime in Docker Container

ENV MULE_HOME /opt/mule


ADD mule-ee-distribution-standalone-4.2.2.zip /opt

ADD mule-hello.jar /opt


RUN set -x \

                                && cd /opt \

                                && unzip mule-ee-distribution-standalone-4.2.2.zip \

                                && mv mule-enterprise-standalone-4.2.2 mule
```

```
WORKDIR $MULE_HOME
VOLUME $MULE_HOME/apps
VOLUME $MULE_HOME/conf
VOLUME $MULE_HOME/domains
VOLUME $MULE_HOME/logs

# Copy and install license
CMD echo "------ Copy and install license --------"
COPY        muleLicenseKey.lic $MULE_HOME/conf/
#RUN $MULE_HOME/bin/mule -installLicense $MULE_HOME/conf/muleLicenseKey.lic

#Check if Mule Licence installed
#RUN ls -ltr $MULE_HOME/conf/
#CMD echo "------ Licence installed ! --------"

#Copy and deploy mule application in runtime
CMD echo "------ Deploying mule application in runtime ! --------"
COPY  mule-hello.jar $MULE_HOME/apps/
RUN ls -ltr $MULE_HOME/apps/

# HTTP Service Port
# Expose the necessary port ranges as required by the Mule Apps
EXPOSE      8082-8091
EXPOSE      9000
EXPOSE      9082

# Mule remote debugger
EXPOSE      5000
# Mule JMX port (must match Mule config file)
EXPOSE      1098
# Mule MMC agent port
EXPOSE      7777
# AMC agent port
EXPOSE      9997
# Start Mule runtime

# Start Mule runtime
CMD echo "------ Start Mule runtime --------"
ENTRYPOINT ["./bin/mule"]
```

5. We are now ready to build the image in docker. Open the command prompt and type the following command inside the PoC folder:

```
docker build -t uhgdockermule -f DockerFile .
```

uhgdockermule is the image name. You can change it as per need.

Output of the build command:

DockerBuildLogs.log

6. Run the docker:

```
docker run -p 8082:8091 -t -i uhgdockermule
```

8091 port will be now mapped to 8082 port

Output:

DockerRunlogs.log

7. Test the deployed application:

a) use docker ps to collect Name as shown:

Sample output:

DockerPS.log

b) use “docker exec -it upbeat_greider /bin/bash”

Sample output:

DockerExec.log

Highlighted is the Name obtained from running instance of Docker container from previous step.

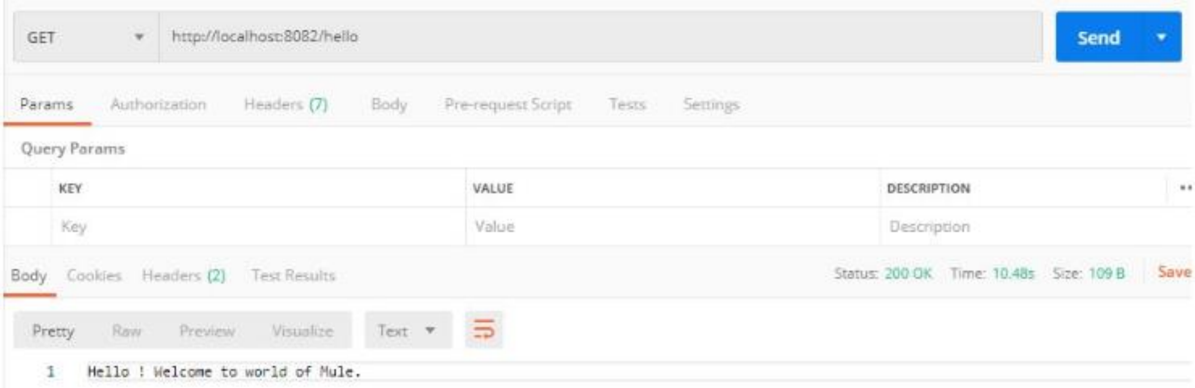
```
root@9eaaa29b363a:/opt/mule/apps# ls -ltr

total 8

drwxr-xr-x 4 root root 4096 Mar  6 00:16 mule-hello

-rw-r--r-- 1 root root   77 Mar  6 00:18 mule-hello-anchor.txt
```

c) Test it with POSTMAN:



GET http://localhost:8082/hello

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 200 OK Time: 10.48s Size: 109 B

```
Hello ! Welcome to world of Mule.
```

Logs:

DockerPostManTest.log