Create an Odysseus Docker Container

In this guide we will show you how to create a Docker container for Odysseus. Such a container can help you to run and share Odysseus more easily. If you want to use our predefined Odysseus Server Docker Container, see Run with Docker.

Step-by-step guide

We assume that you have a Linux system up and running, e.g. Ubuntu. If you want to try it out without changing your system or if you don't have a Linux machine, we recommend VirtualBox.

Installing Docker

1. Open a Terminal window.
2. Type "docker"
3. If Docker is already installed, a help page with common Docker options and commands will show up.
4. If not, your Terminal will probably give you a hint how to install Docker. Under Ubuntu, it is "sudo apt-get install docker.io"

Preparing the Folder

Now we need a folder where we can put everything that we need in.

1. Create a folder, e.g. in your "Documents"-folder.
2. Move to that folder
3. Create another folder "odysseus" in that folder

Creating the magic folder

```
tobi@tobi-VirtualBox:~$ ls
Desktop  Documents  Downloads  examples.desktop  Music  Pictures  Public  Templates  Videos
tobi@tobi-VirtualBox:~$ cd Documents/
tobi@tobi-VirtualBox:~/Documents$ ls
tobi@tobi-VirtualBox:~/Documents$ mkdir odysseusdocker
tobi@tobi-VirtualBox:~/Documents$ ls
odysseusdocker
tobi@tobi-VirtualBox:~/Documents$ cd odysseusdocker/
tobi@tobi-VirtualBox:~/Documents/odysseusdocker$ mkdir odysseus
```

1. Now we want to create the Dockerfile which describes our Docker container
2. Create a new file called "Dockerfile" within our "odysseusdocker"-folder (e.g. with nano again or whatever you prefer)
3. Put the content from the box below into the Dockerfile

Dockerfile

```
FROM ubuntu
MAINTAINER <Your Name>
LABEL Description="Docker Image for running the Odysseus data stream management system" Version="0.1"  URL="
http://odysseus.informatik.uni-oldenburg.de"

# Install required packages
RUN apt-get -y update && apt-get -y install apt-utils wget unzip openjdk-8-jre bash nano

# Download nightly build
RUN mkdir -p /usr/lib/odysseus && mkdir -p /var/log/odysseus
RUN wget -c http://odysseus.offis.uni-oldenburg.de/download/products/server/lastBuild/odysseus.server-linux.gtk.
x86_64.zip -O /tmp/odysseus.zip && unzip /tmp/odysseus.zip -d /usr/lib/odysseus && rm -f /tmp/odysseus.zip

# Define additional VM parameters
RUN echo "-XX:NativeMemoryTracking=summary" >> /usr/lib/odysseus/odysseus.ini && \
    echo "-XX:+HeapDumpOnOutOfMemoryError" >> /usr/lib/odysseus/odysseus.ini && \
    echo "-XX:HeapDumpPath=/var/log/odysseus/dumps" >> /usr/lib/odysseus/odysseus.ini && \
    echo "-XX:ErrorFile=/var/log/odysseus/dumps/hs_err_pid%p.log" >> /usr/lib/odysseus/odysseus.ini && \
    sed -i "s/org\.apache\.log4j\.ConsoleAppender/org\.apache\.log4j\.FileAppender\nlog4j\.appender\.default\.
File=\/var\/log\/odysseus\/odysseus\.log/" /usr/lib/odysseus/plugins/de.uniol.inf.is.odysseus.slf4j_1.7.16
/log4j.properties

# Create group and user
RUN groupadd odysseus && \
    useradd -r -g odysseus -s /bin/bash -d /var/lib/odysseus -m odysseus && \
    chown -R odysseus /var/log/odysseus

# Change timezone for UTC
RUN echo "UTC" > /etc/timezone

# Expose web server port for wsdl
EXPOSE 9669

# Change user and setup environment
USER odysseus
ENV home /var/lib/odysseus
WORKDIR /var/lib/odysseus

CMD ["/usr/lib/odysseus/odysseus"]
```

Now we have everything prepared in that folder.

Build and run the Docker container

1. In your Terminal window, navigate to the "odysseusdocker"-folder
2. Type "sudo docker build ." (The " ." refers to the folder we are in. The command assumes a file named "Dockerfile" in that folder.)
3. Wait (it will take some time to get the basis for our container: Linux and OpenJDK Java 8. But next time you do it it's much faster.)
4. Now we should see a message that the container was successfully build. We can type "sudo docker images" to see our images.

Our Docker images

```
Successfully built c39f4bd5fd33
tobi@tobi-VirtualBox:~/Documents/odysseusdocker$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
<none>              <none>              c39f4bd5fd33        2 minutes ago       686.8 MB
java                8                   c518da75d9f0        6 days ago          642.9 MB
```

Now we can run the Docker container:

1. "sudo docker run -p 127.0.0.1:9700:9669 --name our_odysseus -t c39f4bd5fd33"

Starting the container

```
tobi@tobi-VirtualBox:~/Documents/odysseusdocker$ sudo docker run -p 127.0.0.1:9700:9669 --name our_odysseus -t
c39f4bd5fd33
Starting Odysseus ...
Odysseus: Cannot open display:
Odysseus:
GTK+ Version Check
0    INFO  Core  - Current size of heap:     1.0 GB - de.uniol.inf.is.odysseus.core.Activator.start(Activator.
java:61)
35   INFO  Core  - Maximum size of heap:     1.0 GB - de.uniol.inf.is.odysseus.core.Activator.start(Activator.
java:64)
41   INFO  Core  - Free memory of the heap:     879.4 MB - de.uniol.inf.is.odysseus.core.Activator.start
(Activator.java:67)
52   INFO  Core  - Running VM with:     amd64 - de.uniol.inf.is.odysseus.core.Activator.start(Activator.java:
70)
1912 INFO  OdysseusConfiguration  - Read config file from /root/.odysseus/odysseus.conf - de.uniol.inf.is.
odysseus.core.server.OdysseusConfiguration.loadProperties(OdysseusConfiguration.java:86)
4561 WARN  LogicalOperatorBuilder  - URL for SetTimeProgressMarker not available! - de.uniol.inf.is.odysseus.
logicaloperator.LogicalOperatorBuilder.addLogicalOperator(LogicalOperatorBuilder.java:243)
5699 INFO  OdysseusServerApplication  - Odysseus application is fully started! - de.uniol.inf.is.odysseus.
product.server.starter.OdysseusServerApplication.start(OdysseusServerApplication.java:31)
osgi> 9005 INFO  SingleSchedulerManager  - No Scheduler-Config-File found. - de.uniol.inf.is.odysseus.scheduler.
manager.singleschedulermanager.SingleSchedulerManager.activate(SingleSchedulerManager.java:113)
9030 INFO  SingleSchedulerManager  - New Scheduler-Config-File created - de.uniol.inf.is.odysseus.scheduler.
manager.singleschedulermanager.SingleSchedulerManager.activate(SingleSchedulerManager.java:126)
9109 INFO  SingleSchedulerManager  - Active scheduler. class de.uniol.inf.is.odysseus.scheduler.
singlethreadscheduler.SimpleThreadSchedulerMSLimitSourcesPerThread - de.uniol.inf.is.odysseus.scheduler.manager.
singleschedulermanager.SingleSchedulerManager.activate(SingleSchedulerManager.java:137)
13453 INFO  WebServicePublisher  - Webservice published at http://0:0:0:0:0:0:0:0:9669/odysseus - de.uniol.inf.
is.odysseus.planmanagement.executor.webservice.server.WebServicePublisher.publish(WebServicePublisher.java:106)
```

As you can see above, Odysseus started. We also exposed one port the the outside: port 9669 is forwarded to port 9700, wherefore we should be able to see the WebService in our host machine. Just open http://localhost:9700/odysseus?wsdl in a browser on your host machine.You should be able to see an XML-file describing the Odysseus WebService.

Push the container to DockerHub

We want to make is easy for others to install Odysseus. Hence, we will push our image to DockerHub.

1. First, we need to tag our container: sudo docker tag c39f4bd5fd33 tobi42/odysseus_server:03-16
    a. Notice, that the name "tobi42" before the slash is your account name on Dockerhub, next is the image name and after the : you have the version label or tag
2. Next, we login to DockerHub: sudo docker login
3. Now push it to DockerHub: sudo docker push tobi42/odysseus_server:03-16
4. Take a look on your DockerHub account. You should be able to see your container.

Try out to pull our container

Now we want to try our if our docker container works as expected. First, we need to remove everything we did until now. If we don't do this, docker won't pull the image because it is the same as the one we pushed in the step before.

1. sudo docker stop our_odysseus
2. sudo docker rm our_odysseus
3. sudo docker rmi $(sudo docker images -q)

Now, everything we did should be removed. Let's run Odysseus from DockerHub.

1. sudo docker run -p 127.0.0.1:9700:9669 --name our_odysseus tobi42/odysseus_server:03-16

Now, Odysseus should run exactly as it was running in our previous local test. You should be able to see the WebService description in your browser: http://localhost:9700/odysseus?wsdl

Some Docker management

- See all your running containers: sudo docker ps
- See all your started containers (including non-running containers): sudo docker ps -a
- Stop all running containers: sudo docker stop $(sudo docker ps -a -q)
- Delete all started containers: sudo docker rm $(sudo docker ps -a -q)
- Delete all images: `sudo docker rmi $(sudo docker images -q)`