Docker (<https://www.docker.com/>) with OpenCPU

Docker is available in two editions: **Community Edition (CE)** and **Enterprise Edition (EE)**.

Docker Community Edition (CE) is ideal for developers and small teams looking to get started with Docker and experimenting with container-based apps. Docker CE has two update channels, **stable** and **edge**:

* **Stable** gives you reliable updates every quarter
* **Edge** gives you new features every month

For our purposes, the stable Docker Community Edition (CE) is used.

1. Install docker
   1. For Windows from <https://store.docker.com/editions/community/docker-ce-desktop-windows>
   2. For Mac from <https://www.docker.com/docker-mac>

A whale should be popping up in the taskbar (and “Docker is running”). By right-click, you have some options. If you have no internet, connection (e.g. WLAN) use a good docking station with LAN cable for internet access.

Let assume that we are on Windows! Otherwise you should set “sudo” always before the docker command in the terminal.

1. Test docker in Windows PowerShell: docker run hello-world
2. Search for an Ubuntu image on <https://hub.docker.com/_/ubuntu/>
3. Create a Dockerfile (by using in PowerShell: New-Item C:\...\openCPU\Dockerfile –type file)
4. Type the commands in this file as follows (lines beginning with # are comments):
5. Always begin with FROM Ubuntu: <version>

Example:

# Use builds from launchpad

FROM ubuntu:16.04

ENV DEBIAN\_FRONTEND noninteractive

RUN \

apt-get update && \

apt-get -y dist-upgrade && \

apt-get install -y software-properties-common && \

add-apt-repository -y ppa:opencpu/opencpu-2.0 && \

apt-get update && \

apt-get install -y opencpu-server

# Prints apache logs to stdout

RUN \

ln -sf /proc/self/fd/1 /var/log/apache2/access.log && \

ln -sf /proc/self/fd/1 /var/log/apache2/error.log && \

ln -sf /proc/self/fd/1 /var/log/opencpu/apache\_access.log && \

ln -sf /proc/self/fd/1 /var/log/opencpu/apache\_error.log

# Set opencpu password so that we can login

RUN \

echo "opencpu:opencpu" | chpasswd

1. If you have to install package from CRAN you should install wget as follows:

RUN apt-get install -y wget

1. Copy your own R-packages:

# Install R package

COPY digiterEmpty\_0.1.0.tar.gz /tmp

COPY digiterSmall\_0.1.1.tar.gz /tmp

COPY digiterLarge\_0.1.1.tar.gz /tmp

1. Install all wanted packages

RUN wget https://cran.r-project.org/src/contrib/randomForest\_4.6-12.tar.gz -P /tmp

RUN R CMD INSTALL /tmp/randomForest\_4.6-12.tar.gz --library=/usr/local/lib/R/site-library

RUN R CMD INSTALL /tmp/digiterEmpty\_0.1.0.tar.gz --library=/usr/local/lib/R/site-library

RUN R CMD INSTALL /tmp/digiterSmall\_0.1.1.tar.gz --library=/usr/local/lib/R/site-library

RUN R CMD INSTALL /tmp/digiterLarge\_0.1.1.tar.gz --library=/usr/local/lib/R/site-library

1. Optional: Load packages by building the docker image with:

RUN sed -i 's/\"lattice\"/\"lattice\",\"randomForest\", \"digiterEmpty\", \"digiterSmall\", \"digiterLarge\"/' /etc/opencpu/server.conf

Therefore, the lattice package should be preinstalled int the Ubuntu-image from <https://hub.docker.com/_/ubuntu/> .

1. Expose the using port:

# Apache ports

EXPOSE 80

1. End with the command:

# Start non-daemonized webserver

CMD apachectl -DFOREGROUND

The whole example looks like this:

# Use builds from launchpad

FROM ubuntu:16.04

ENV DEBIAN\_FRONTEND noninteractive

RUN \

apt-get update && \

apt-get -y dist-upgrade && \

apt-get install -y software-properties-common && \

add-apt-repository -y ppa:opencpu/opencpu-2.0 && \

apt-get update && \

apt-get install -y opencpu-server

# Prints apache logs to stdout

RUN \

ln -sf /proc/self/fd/1 /var/log/apache2/access.log && \

ln -sf /proc/self/fd/1 /var/log/apache2/error.log && \

ln -sf /proc/self/fd/1 /var/log/opencpu/apache\_access.log && \

ln -sf /proc/self/fd/1 /var/log/opencpu/apache\_error.log

# Set opencpu password so that we can login

RUN \

echo "opencpu:opencpu" | chpasswd

RUN apt-get install -y wget

# Install R package

COPY digiterEmpty\_0.1.0.tar.gz /tmp

COPY digiterSmall\_0.1.1.tar.gz /tmp

COPY digiterLarge\_0.1.1.tar.gz /tmp

RUN wget https://cran.r-project.org/src/contrib/randomForest\_4.6-12.tar.gz -P /tmp

RUN R CMD INSTALL /tmp/randomForest\_4.6-12.tar.gz --library=/usr/local/lib/R/site-library

RUN R CMD INSTALL /tmp/digiterEmpty\_0.1.0.tar.gz --library=/usr/local/lib/R/site-library

RUN R CMD INSTALL /tmp/digiterSmall\_0.1.1.tar.gz --library=/usr/local/lib/R/site-library

RUN R CMD INSTALL /tmp/digiterLarge\_0.1.1.tar.gz --library=/usr/local/lib/R/site-library

RUN sed -i 's/\"lattice\"/\"lattice\",\"randomForest\", \"digiterEmpty\", \"digiterSmall\", \"digiterLarge\"/' /etc/opencpu/server.conf

# Apache ports

EXPOSE 80

# Start non-daemonized webserver

CMD apachectl –DFOREGROUND

1. Put the .tar.tz file into the same directory as the Dockerfile
2. In PowerShell:
3. Set the directory from the dockerfile by cd C:\...\openCPU
4. docker build . (This gives the image ID as “Successfully built 9f6825b856aa”). So in this example <image ID>=9f6825b856aa .
5. docker run -p 80:80 --name <new image name> <image ID> e.g. docker run -p 80:80 --name opencpu 9f6825b856aa
6. If it stated “OpenCPU cloud server ready”, one can test the port and make GET/POST requests. You can make the requests using R directly, using Postman (<https://www.getpostman.com/> ), using Python or some other languages.

The url should look like this:

Local:

<http://localhost:port_number/ocpu/library/package_name/R/package_function/json>

On a virtual machine:

http://lin-op-vm.westeurope.cloudapp.azure.com:port\_number/ocpu/library/package\_name/R/package\_function/json

Example:

http://localhost:80/ocpu/library/digiterSmall/R/predict\_digit\_small/json

http://lin-op-vm.westeurope.cloudapp.azure.com:80/ocpu/library/digiterEmpty/R/predict\_digit\_empty/json