nlmixr docker installation

Richard Hooijmaijers

9/27/2018

Docker

The first thing to do is install docker. An installer is available for all major operating systems:

- windows: https://docs.docker.com/docker-for-windows/install/
- mac: https://docs.docker.com/docker-for-mac/install/
- ubuntu: https://docs.docker.com/install/linux/docker-ce/ubuntu/

Make sure you read the prequisites, **especially on windows this is not always straightforward**. On other systems it is sufficient to download and run the installer.

Running the container

To run a docker container you should use a terminal window (or command prompt in windows). The first time a container is ran, it will donwload all applicable files from the repository. This can take some time depending on your internet speed. After the first time, the container will be saved localy and it the machine will start almost immediately. For working with the packages, it will be necessary to have access to your local drive. This can be done by adding an option (-v) in the docker command. In the following example the folder "Users/richard/Document" (this could be any folder that is accessible on your system) will be mounted to the docs folder in the home directory:

```
docker run -v /Users/richard/Documents/:/home/rstudio/docs -d -p 8787:8787 nlmixr/nlmixr:V1.3
```

On windows this can take some more effort. The following method was proposed by Nick Holford (link):

- Go to Docker settings shared drives and enter a local drive letter e.g. C. . This then requires providing login credentials for your local device to validate the choice of drive. That requirement could mean setting up a local account on Windows instead of the Microsoft account.
- Enter the drive letter and a colon e.g. C: before the path to the local directories. The trailing path separator is optional. It works with or without the "/".
- Including the /home/rstudio/docs path is essential in order to have access to the local directories (in /home/rstudio/docs).

```
docker run -v C:/Users/nholf/Documents:/home/rstudio/docs -d -p 8787:8787 nlmixr/nlmixr:1.3
```

You now have access to other directories in /home/rstudio in addition to examples and kitematic (i.e. analysis, data, models, scripts, ShinyMixR).

Once the docker container is running, you can go to a browser and go to the following url:

```
localhost:8787
```

This will fire up an Rstudio server version. If everything went correctly you will see a login screen for Rstudio. You can login with username and password **rstudio**. If a local folder was mounted you should be able to find this directly in the files pane in Rstudio.

Note: In certain cases the localhost is not recognized and the url should be `127.0.0.1:8787`

Note: If you are using Docker Toolbox on Windows 7, use the Docker Machine IP instead of localhost. For example, http://192.168.99.100:8787/. To find the IP address, the `ipconfig` command can be used.

Note: In general most modern browsers will work with rstudio server. However there might be some problems when running shiny apps. Both chrome and safari were tested and should work fine.

Test the installation

It is possible to test if all packages are running correctly by downloading and running the test script:

```
# install.packages("RCurl")
library(RCurl)
url <- "https://raw.githubusercontent.com/nlmixrdevelopment/nlmixr/master/build/test_install.R"
script <- getURL(url)
eval(parse(text = script))
```

Stop the container

Once you are done, or you would want to start another instance of the container, you can close the browser window and from the terminal the following can be done to stop all containers:

```
docker kill $(docker ps -q)
```

Each time you stop the container or restart/log-off, the container should be started again using the `docker run` command as stated above.

Note: In case you want to stop a specific container, you can list the running containers with `docker container ls` and close the specific container ID (e.g. `docker container stop 1fa4ab2cf395`)

Note: In windows it might be necessary to stop the container using powershell, otherwise the container can be stopped using task manager