



udocker - *be anywhere*

Part 3 - Hands On: intermediate stuff

<https://github.com/indigo-dc/udocker>

Mario David david@lip.pt, Jorge Gomes jorge@lip.pt



Importing and exporting, loading and saving: images and containers

I have a dockerfile!

- *But udocker does not support `build` the dockerfile...*
 - Use `docker` itself in you <lap|desk>top
 - Example: <https://github.com/mariojmdavid/docker-gromacs-cuda/blob/master/gromacs-cpu/Dockerfile-cpu>

```
git clone https://github.com/mariojmdavid/docker-gromacs-cuda.git
cd docker-gromacs-cuda/gromacs-cpu/
docker build --build-arg gromacs_ver=2023 -t gromacs -f Dockerfile-cpu .
```

- (Will take quite awhile)

I have a docker image!

After you build the image with docker:

```
docker images
REPOSITORY    TAG          IMAGE ID      CREATED       SIZE
gromacs       latest      8473080f1963 3 minutes ago 376MB
```

Save the image with `docker` to a tarball:

```
docker save -o gromacs.tar gromacs
```

udocker load

You can load a tarball with udocker that is a docker image, and that you saved previously with docker:

```
udocker load -i gromacs.tar gromacs
```

And now you can check several things:

```
udocker images  
REPOSITORY  
gromacs:latest
```

Create a container and run it

```
udoocker create --name=grom gromacs
```

```
udoocker ps
```

CONTAINER ID	P	M	NAMES	IMAGE
e2e014d9-9770-3fb5-a4a9-098a95371adf	.	W	['grom']	gromacs:latest

```
udoocker run grom env
```

```
*****  
*                                                                 *  
*           STARTING e2e014d9-9770-3fb5-a4a9-098a95371adf       *  
*                                                                 *  
*****  
executing: env  
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/gromacs/bin  
LD_LIBRARY_PATH=:/usr/local/gromacs/lib
```

Running gromacs with udocker

```
udocker run grom gmx mdrun -h
*****
executing: gmx
                :-) GROMACS - gmx mdrun, 2022 (-:
Executable:    /usr/local/gromacs/bin/gmx
Data prefix:   /usr/local/gromacs
Working dir:   /home
Command line:
    gmx mdrun -h
SYNOPSIS
gmx mdrun [-s [<.tpr>]] [-cpi [<.cpt>]] [-table [<.xvg>]] [-tablep [<.xvg>]]
```

Environment in dockerfile is preserved - I

You can check the dockerfile: <https://github.com/mariojmdavid/docker-gromacs-cuda/blob/master/gromacs-cpu/Dockerfile-cpu>

```
FROM ubuntu:22.04
LABEL maintainer="Mario David <mariojmdavid@gmail.com>"
...
ENV PATH=$PATH:/usr/local/gromacs/bin
ENV LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/gromacs/lib
WORKDIR /home
```


Environment in dockerfile is preserved in udocker container - II

Just check the `ENV` and `WORKDIR` :

```
udocker run grom env
...
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/gromacs/bin
LD_LIBRARY_PATH=:/usr/local/gromacs/lib

udocker run grom pwd
...
/home
```

I want to install/compile in a container! - I

Pull some base image, create a container and run:

```
udocker pull almalinux:9  
udocker create --name=mypython almalinux:9  
udocker run mypython bash
```

And after that install and/or compile whatever you want

I want to install/compile in a container! - II

Now you are inside the container and seems you are `root`:

```
dnf -y install python3 gcc-c++ python3-pip  
pip-3 install numpy matplotlib scipy  
exit
```

You are satisfied so you exit the container, but... I want to preserve what I installed.

udocker export and import

You can export a container into a tarball, for safekeeping:

```
udocker export -o mypython.tar mypython
```

Now you can import this container into an image with a given tag (empty tag defaults to `latest`):

```
udocker import mypython.tar mypython:v1.0
udocker images
REPOSITORY
...
mypython:v1.0
```

About mounting volumes and directories

Mounting a directory in the container - I

Assume you have a directory you want to use inside the container, and grab yourself a tpr file:

```
mkdir -p $HOME/udocker-tutorial/gromacs/input  
cd $HOME/udocker-tutorial/gromacs/input/  
wget --no-check-certificate https://download.ncg.ingrid.pt/webdav/gromacs-input/md.tpr
```

Mounting a directory in the container - II

We will bind mount the directory in the `/home/user` inside the container (if this directory does not exist inside the container, then it will be created):

```
udocker run -v=$HOME/udocker-tutorial/gromacs:/home/user -w=/home/user grom /bin/bash
```

Mounting a directory in the container - III

Now, inside the container:

```
ls -al
total 12
drwxrwxr-x 3 root root 4096 Apr  4 08:31 .
drwxr-xr-x 3 root root 4096 Apr  4 08:42 ..
drwxrwxr-x 2 root root 4096 Apr  4 08:31 input
```


Mounting a directory in the container - IV

Inside the container - make a directory for your output, and run your favorite molecular dynamics simulation (if you want wait a few minutes to finish, will not take long):

```
mkdir output
cd output
gmx mdrun -s /home/user/input/md.tpr -deffnm ud-tutorial \
    -maxh 0.50 -rethway -noconfout -nsteps 10000 -g logfile

exit
```

Mounting a directory in the container - V

And back to your preferred machine:

```
ls $HOME/udocker-tutorial/gromacs/output  
logile.log  ud-tutorial.edr  ud-tutorial.trr  ud-tutorial.xtc
```

All nice output files right there in you home directory.

End of Hands On part II

