udocker - *be anywhere*

Part 2 - Hands On: basic stuff

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

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

What udocker is not - I

- Not appropriate to run services:
  - In most cases you need root privileges to run services.
  - You have Docker (or other container tools) for this.
- udocker is a run-time and is not meant to build docker images:
  - Docker images should be built with Docker.
  - Use you (Lap/Des)top with Docker, for this.

What udocker is not - II

- docker-compose like functionality:
  - This is usually to compose micro-services to deploy a platform/service.
  - Again udocker is not appropriate to run services.
  - Use docker-compose itself for this.

udocker aims/objectives

- Execute applications encapsulated with dependencies in containers
  - as non privilege user.
- Execute containers from docker images
  - including officially supported images in Dockerhub.
- Execute applications with very specific, customized libraries and environments
  - difficult to support in very controlled systems such as HPC machines.

udocker: Installation

https://indigo-dc.github.io/udocker/installation_manual.html

Installation: tarball

Access the INCD advanced computing facility at Lisbon using ssh:

```
ssh -l <username> cirrus8.a.incd.pt
module load python/3.10.8
```

- The end user can download and execute udocker without system administrator intervention.
- Install from a released version:
  - Download a release tarball from https://github.com/indigo-dc/udocker/releases:

```
wget https://github.com/indigo-dc/udocker/releases/download/1.3.10/udocker-1.3.10.tar.gz
tar zxvf udocker-1.3.10.tar.gz
export PATH=`pwd`/udocker-1.3.10/udocker:$PATH
```

Installation: PyPI - I

- Install from PyPI using pip:
  - For installation with pip it is advisable to setup a Python3 virtual environment

```
python3 -m venv udockervenv
source udockervenv/bin/activate
pip install udocker==1.3.10
```

Installation: PyPI - II

The udocker command will be `udockervenv/bin/udocker`.

- Optionally, we can set `UDOCKER_DIR` environment variable where the binaries, libraries images and containers will be saved. The default directory is `$HOME/.udocker`.

```
mkdir udocker-tutorial
cd udocker-tutorial/
export UDOCKER_DIR=$HOME/udocker-tutorial/.udocker
```

(More details: https://indigo-dc.github.io/udocker/installation_manual.html)

Installation: tools and libraries - I

- udocker executes containers using external tools and libraries that are enhanced and packaged for use with udocker.
- To complete the installation, download and install the required tools and libraries.

```
udocker install
```

Installation: tools and libraries - II

- Installs by default in `$HOME/.udocker`, or in `UDOCKER_DIR=$HOME/udocker-tutorial/.udocker`.
- Explore the directory structure under `$HOME/udocker-tutorial/.udocker`

udocker: CLI - the basic (introductory) stuff

https://indigo-dc.github.io/udocker/user_manual.html

0. help and version

Global help and version

```
udocker --help
udocker --version
```

You can get help on a given command

```
udocker run --help
```

1. pull

Pull an image from Dockerhub (for example, an officially supported tensorflow):

```
udocker pull tensorflow/tensorflow
```

2. images

List the images in your local repository (`-l` option shows long format):

```
udocker images
udocker images -l
```

3. create

To create a container named `mytensor`, the default execution engine is P1 (PTRACE + SECCOMP filtering):

```
udocker create --name=mytensor tensorflow/tensorflow
```

4. ps

List extracted containers. These are not processes but containers extracted and available for
execution:

```
udocker ps
```

5. run: I

Executes a container. Several execution engines are provided. The container can be specified using the container id or its associated name. Additionally it is possible to invoke run with an image name:

```
udocker run mytensor bash
```

5. run: II

Now you are inside the container (apparently as `root`), you might as well try out:

```
root@pcdavid:~# python
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
```

Or:

```
udocker run mytensor cat /etc/lsb-release
```

6. setup

With `--execmode` chooses an execution mode to define how a given container will be executed. The option `--nvidia` enables access to NVIDIA GPUs (only possible if they are available).

```
udocker setup --execmode=F1 mytensor
udocker ps -m  # confirm change of execution engine
```

7. rm

Delete a previously created container. Removes the entire directory tree extracted from the container image and associated metadata:

```
udocker rm mytensor
```

8. rmi

Delete a local container image previously pulled/loaded/imported:

```
udocker rmi tensorflow/tensorflow
```

End of Hands On part I