Packaging software in portable manner with Singularity/Apptainer

Olivier Mattelaer
UCLouvain
CP3 & CISM

What do I want to cover

- What is a container
  - ➡ Why it can be interesting for you?

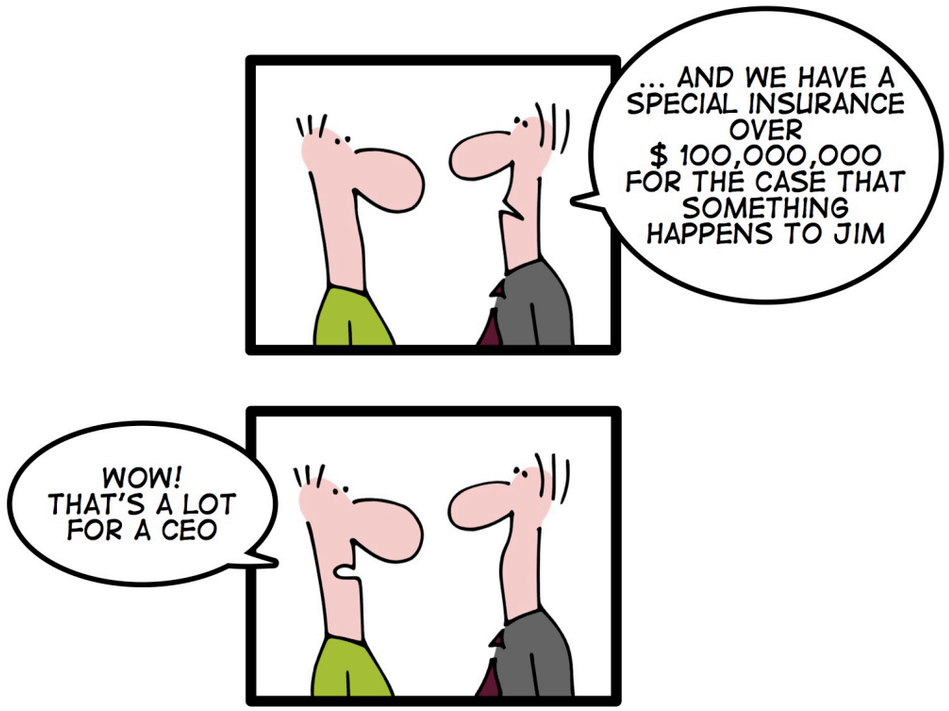
- Singularity/Apptainer: Container for HPC
  - ➡ Features
  - ➡ Limitations

- Small Demo
  - ➡ Show that this is easy to do

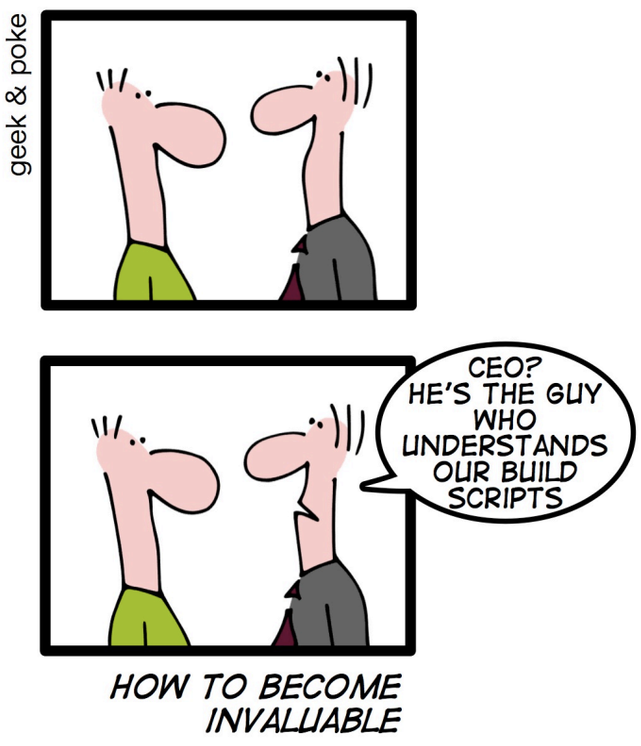
CÉCI

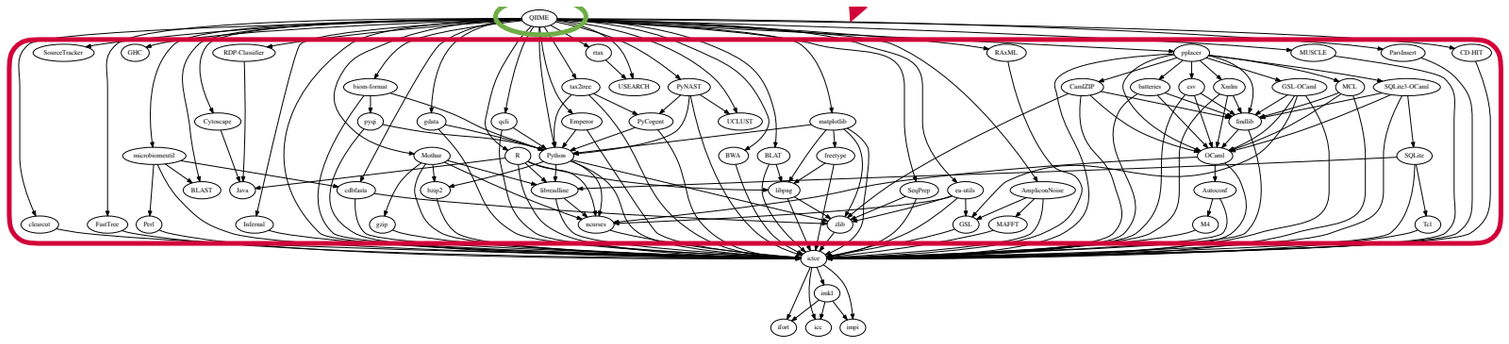
- HPC
  - ➡ Details on how to use our setup

Installing Software

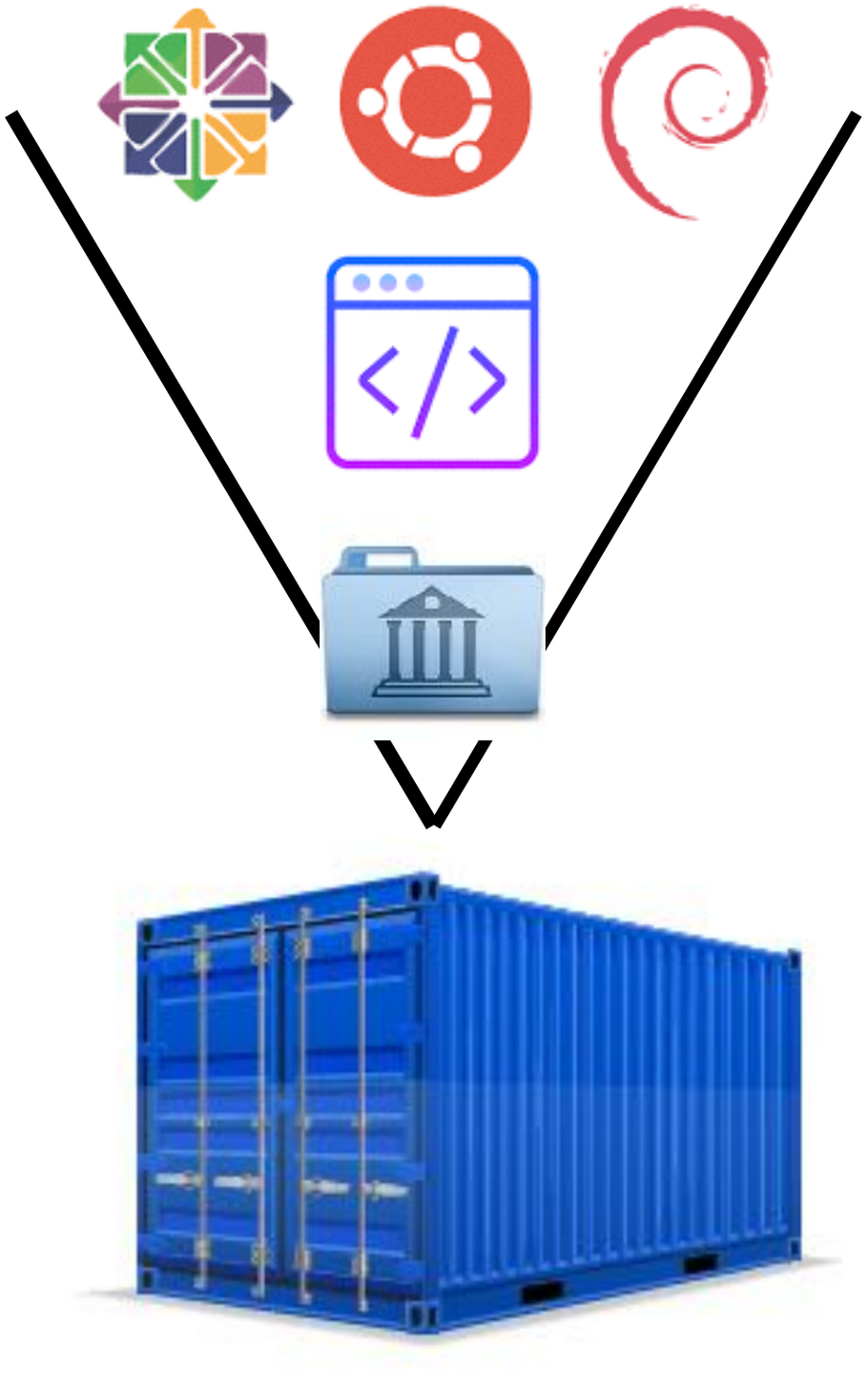


- Tedious/complicated
  - ➡ For user
  - ➡ For sys-admin
- Dependencies Hell



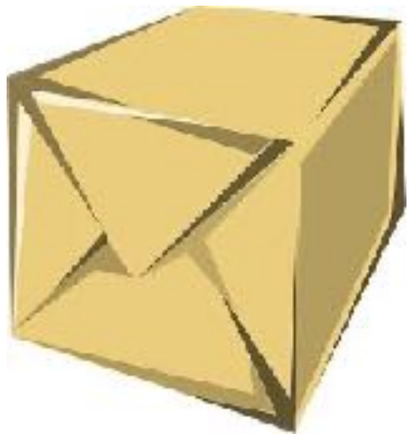


Container Solution



- machine agnostic code
  - ➡ A (small) OS
  - ➡ Your code (executable)
  - ➡ All the dependencies (libraries)
- That can run “everywhere”

What for?

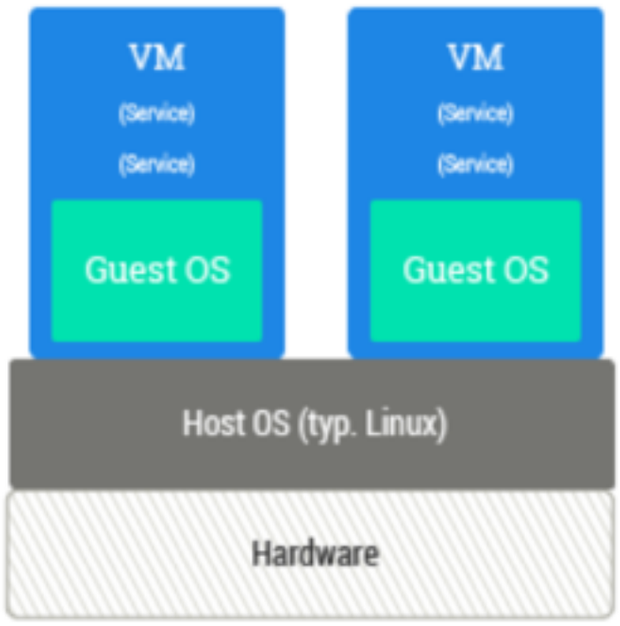


- ➡ **reproducibility** on any (unix) machine
  - ✦ Nice to send to a collaborator !
- ➡ **deployment** (cloud/laptop/hpc/…)
  - ✦ Nice to distribute the workload
  - ✦ Nice for automatic test (CI/CD)
- ➡ With a **paper**
  - ✦ Nice for being able to reproduce results
  - ✦ Nice for other scientists

VM versus container

VM

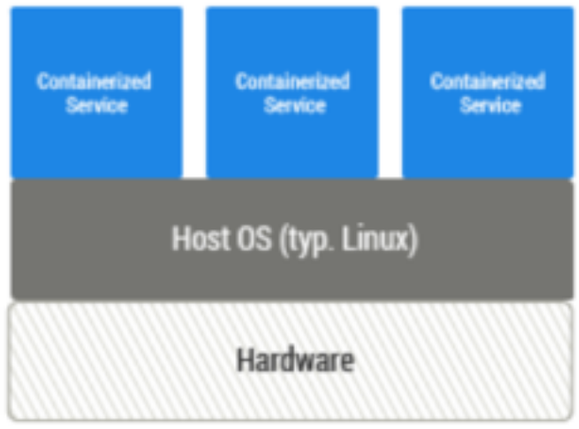
- virtualize the kernel
  - ➡ Hardware virtualisation



- ➡ Flexible
- ➡ slow/resource hungry

container

- Reuse the kernel
  - ➡ Software virtualisation

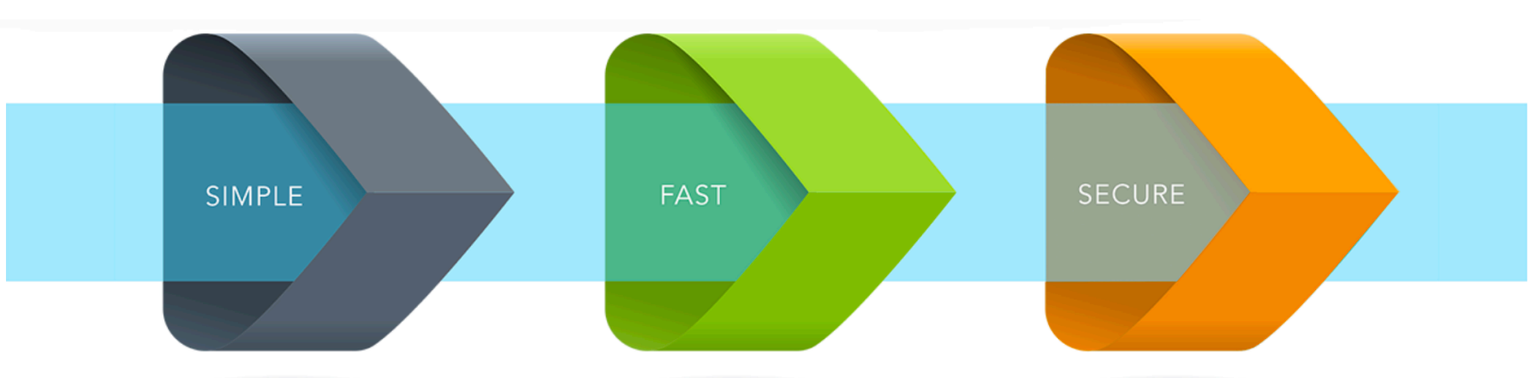


- ➡ Not multi os
- ➡ fast/light
  - ➡ OK for single app
  - ➡ Good for HPC

Containers History

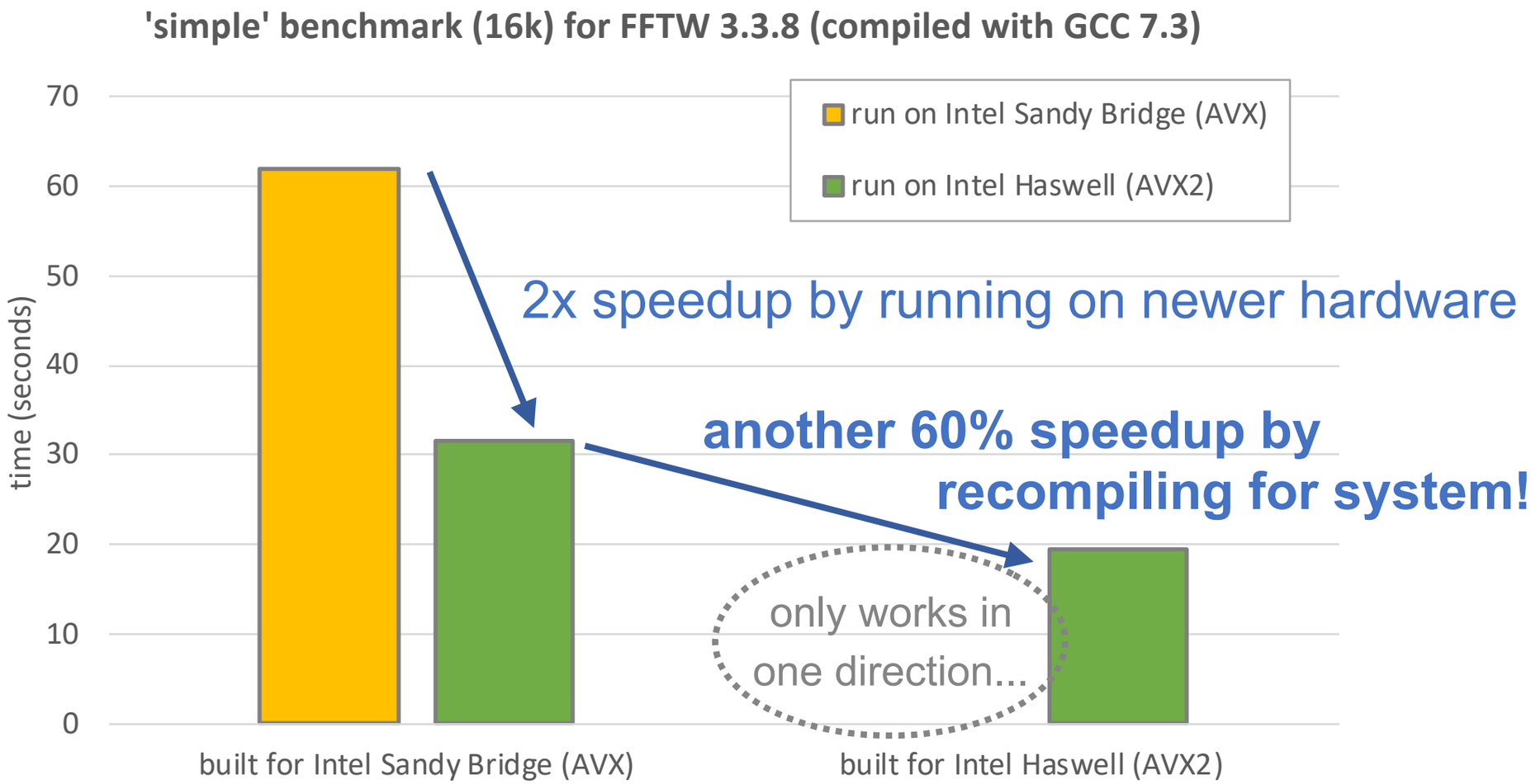
- Are an old idea
  - ➡ Chroot (1979), FreeBSD jails (2000), Solaris containers (2004), LXC (2008)
- Docker (2013)
  - ➡ For/with cloud computing

- Buzz for HPC containers starts ~ 2015
  - ➡ Docker tries to convince HPC structure and failed
- Singularity (2016) and Apptainer (2021)
  - ➡ HPC focus



Performance

- They claim "native" performance
  - ➡ understand "small" overheard (couple of percent)
  - ➡ No cpu optimisation



(FFTW 3.3.8 installed in Singularity container)

Plot taken from Kenneth Hoste

Hardware Optimisation

CPU

Need generic compilation

GPU

Special handling to handle GPU
Specific library at run time

MPI

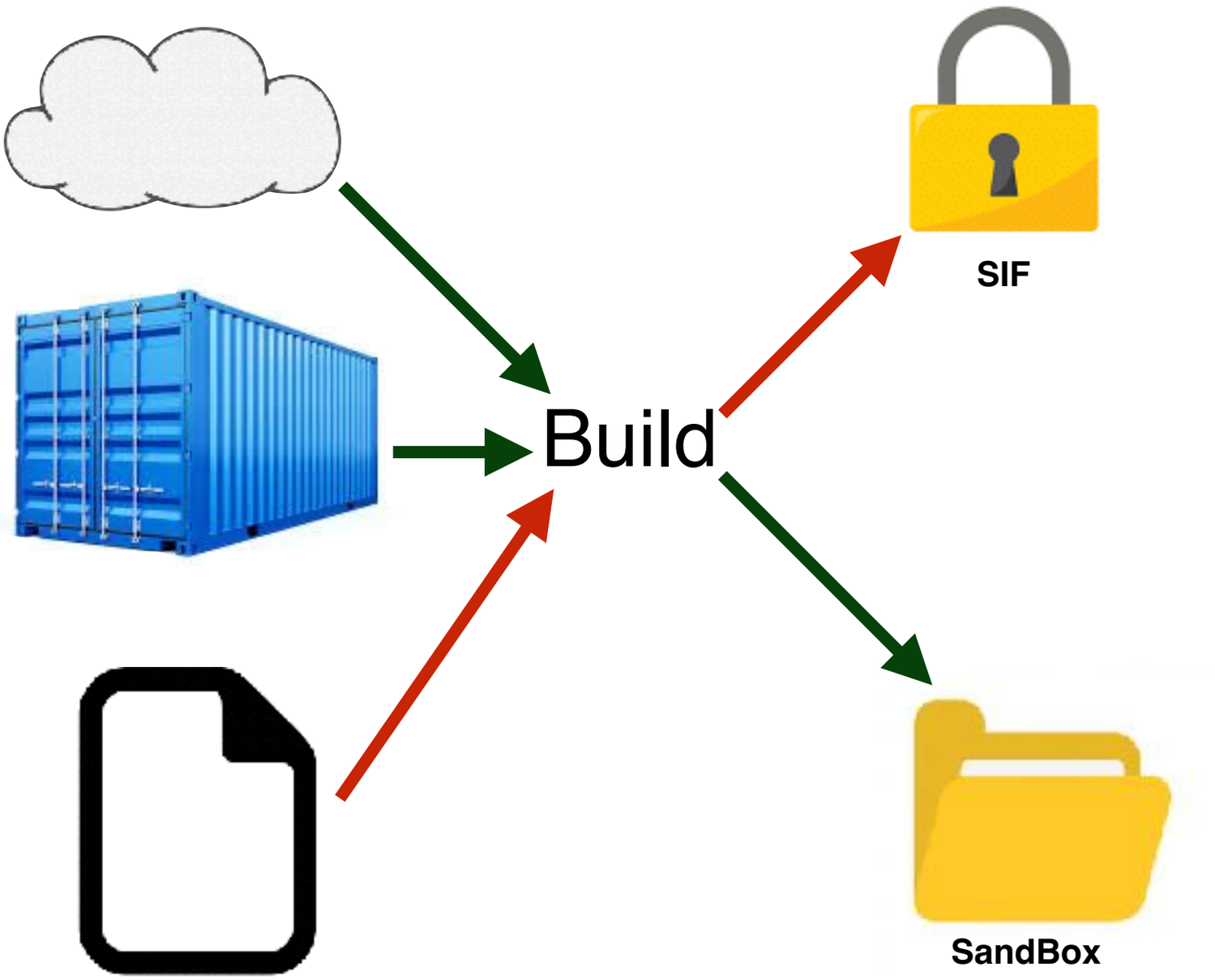
No special handling
But actually needed

No portability here!

https://sylabs.io/guides/3.6/user-guide/mpi.html?highlight=mpi

Building an image

```
$ sudo  singularity build lolcow.simg shub://GodloveD/lolcow
```



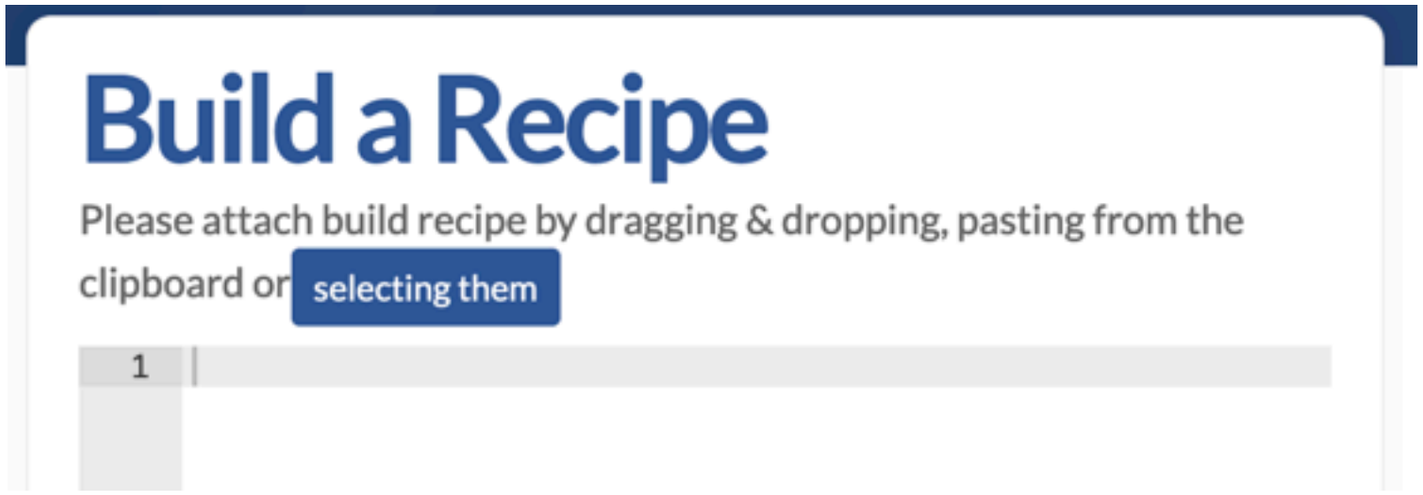
- Singularity Integrity File
  - ➡ Read-only (signed)
  - ➡ default
- Sandbox --sandbox
  - ➡ Full directory
  - ➡ Writable
  - ➡ Can break reproducibility

Root privileges is always required

Remote build

- https://cloud.sylabs.io/home
  - ➡ Allow remote build (No need to be root on your machine)
  - ➡ You can do everything from the CECI clusters
    - ✦ No file transfer

Online



From laptop/cluster

```
[singularity]$ singularity build --remote test_remote.sif shub://Godlove
INFO:    Remote "default" added.
INFO:    Authenticating with remote: default
INFO:    API Key Verified!
INFO:    Remote "default" now in use.
INFO:    Starting build...
 87.57 MiB / 87.57 MiB  100.00% 49.16 MiB/s 1sm01s
INFO:    Creating SIF file...
INFO:    Build complete: /tmp/image-968903817
```

Recipe file

```
Bootstrap: library
From: ubuntu:18.04
```

Based on

```
%runscript
    echo "Container was created $NOW"
    echo "Arguments received: $*"
    exec echo "$@"
```

What to do

```
%post
    apt-get update && apt-get install -y netcat
    NOW=`date`
```

How to install

```
[vagrant@localhost singularity]$ sudo singularity build test.simg centos.def
```

- Other keywords:
  - ➡ files, test, app

Recipe file

```
Bootstrap: library
From: ubuntu:18.04
```

How to start (previous container/…)

```
%setup
    touch /file1
    touch ${SINGULARITY_ROOTFS}/file2
```

Command run on the host

```
%files
    /file1
    /file1 /opt
```

Files copy into the container

```
%environment
    export LISTEN_PORT=12345
    export LC_ALL=C
```

Define environment variables

```
%post
    apt-get update && apt-get install -y netcat
    NOW=`date`
    echo "export NOW=\"${NOW}\"" >> $SINGULARITY_ENVIRONMENT
```

Installation of software within the container

```
%runscript
    echo "Container was created $NOW"
    echo "Arguments received: $*"
    exec echo "$@"
```

Command run via “singularity run”

```
%labels
    Author d@sylabs.io
    Version v0.0.1
```

Information about the container

```
%help
    This is a demo container used to illustrate a def file that uses all
    supported sections.
```

Help about the container

Also %test %startscript + support for app

DEMO

More on filesystem

- Special directory automatically mounted:
  - ➡ $HOME, /tmp, /proc, /sys, /dev
- You can create different mount point
  - ➡ Allow you to specify the path to data/output (specific to system)

```
vagrant@vagrant:~/tuto2$ singularity run --bind /vagrant:/mnt ./hello.simg -i cowcay_now -o /mnt/cowsay_now
This is what happens when you run the container...
vagrant@vagrant:~/tuto2$
```

  - ➡ File is now written in /vagrant of the VM

- Also possible via environment variable:
  - ➡ export SINGULARITY_BINDPATH=/vagrant:/mnt

CÉCI clusters

- Singularity is available on
  - ➡ Lemaitre3
  - ➡ dragon2
  - ➡ Hercules2
  - ➡ Nic5

GPU

- Let's take a image with require some gpu

```
$ singularity pull docker://tensorflow/tensorflow:latest-gpu
...
INFO:    Creating SIF file...
INFO:    Build complete: tensorflow_latest-gpu.sif
```

- To link to the GPU, you need to add —nv

```
$ singularity run --nv tensorflow_latest-gpu.sif


        ________                                _______________
        ___  __/__________________________________  ____/__  /________      __
        __  /  _  _ \_  __ \_  ___/  __ \_  ___/_  /_   __  /_  __ \_ | /| / /
        _  /   /  __/  / / /(__  )/ /_/ /  /   _  __/   _  / / /_/ /_ |/ |/ /
        /_/    \___//_/ /_//____/ \____//_/    /_/      /_/  \____/____/|__/


You are running this container as user with ID 1000 and group 1000,
which should map to the ID and group for your user on the Docker host. Great!

Singularity>
```

MPI

https://support.ceci-hpc.be/doc/_contents/UsingSoftwareAndLibraries/Singularity/index.html

- MPI support requires
  - ➡ That you install the same slurm version as the one on our cluster
  - ➡ That you have the same version of mpi on the machine

- So you need matching pieces
  - ✓ We provide a starting container
    - ➡ Correct version of slurm
    - ➡ For each openmpi version
- You can use such container as base for your work

https://sylabs.io/guides/3.6/user-guide/mpi.html?highlight=mpi

Conclusion

- Singularity
  - ➡ Nice way to share code with colleague
  - ➡ Portability and reproducibility
- Few command to learn
  - ➡ But not that complicated!
- Need to be root on machine
  - ➡ Ok that's annoying…
    - ✦ Virtual machine option quite practical
  - ➡ Remote building exists for recipe files