

Packaging software in portable manner with Singularity

**Olivier Mattelaer
UCLouvain
CP3 & CISM**

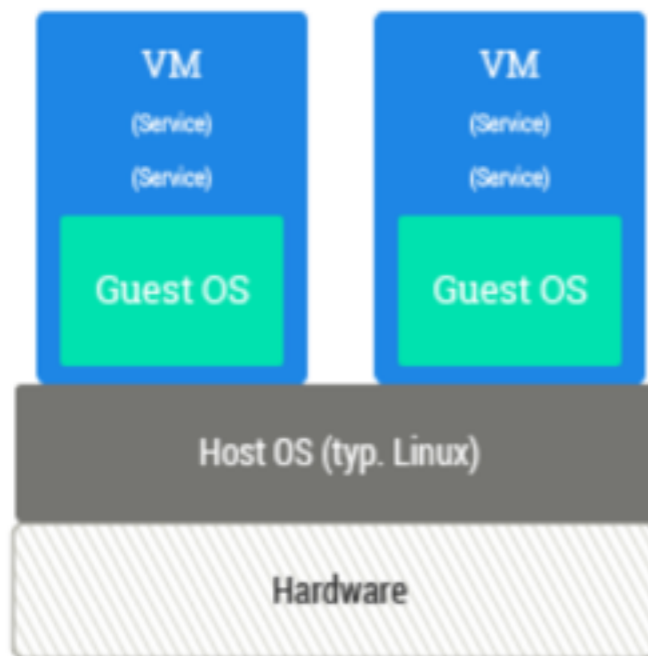
What is a container?

- A package that include
 - ➔ Your application
 - ➔ The dependencies of your application
 - ➔ A (small) os (unix)
- Allow
 - ➔ reproducibility on any (unix) machine
 - ➔ Easy deployment on various architecture (cloud/laptop/hpc/...)
 - ➔ Publish a paper and link it to a container (with input data) such that others can easily reproduce your results.
- Technically:
 - ➔ Code that isolate a code from the rest of the system
 - ◆ Using namespace/cgroup/...

VM versus container

VM

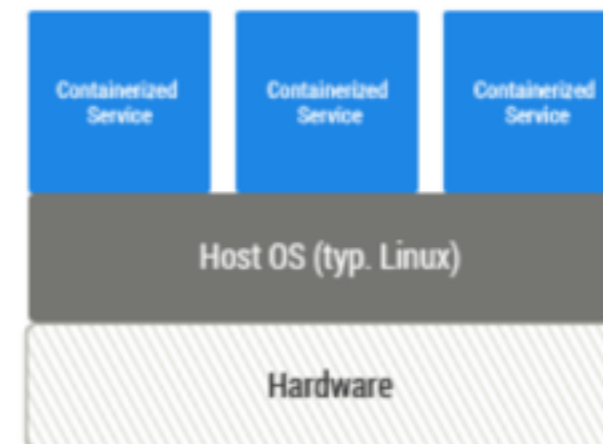
- virtualize the kernel
 - ➔ Hardware virtualisation



- ➔ Flexible
- ➔ slow/ressource hungry

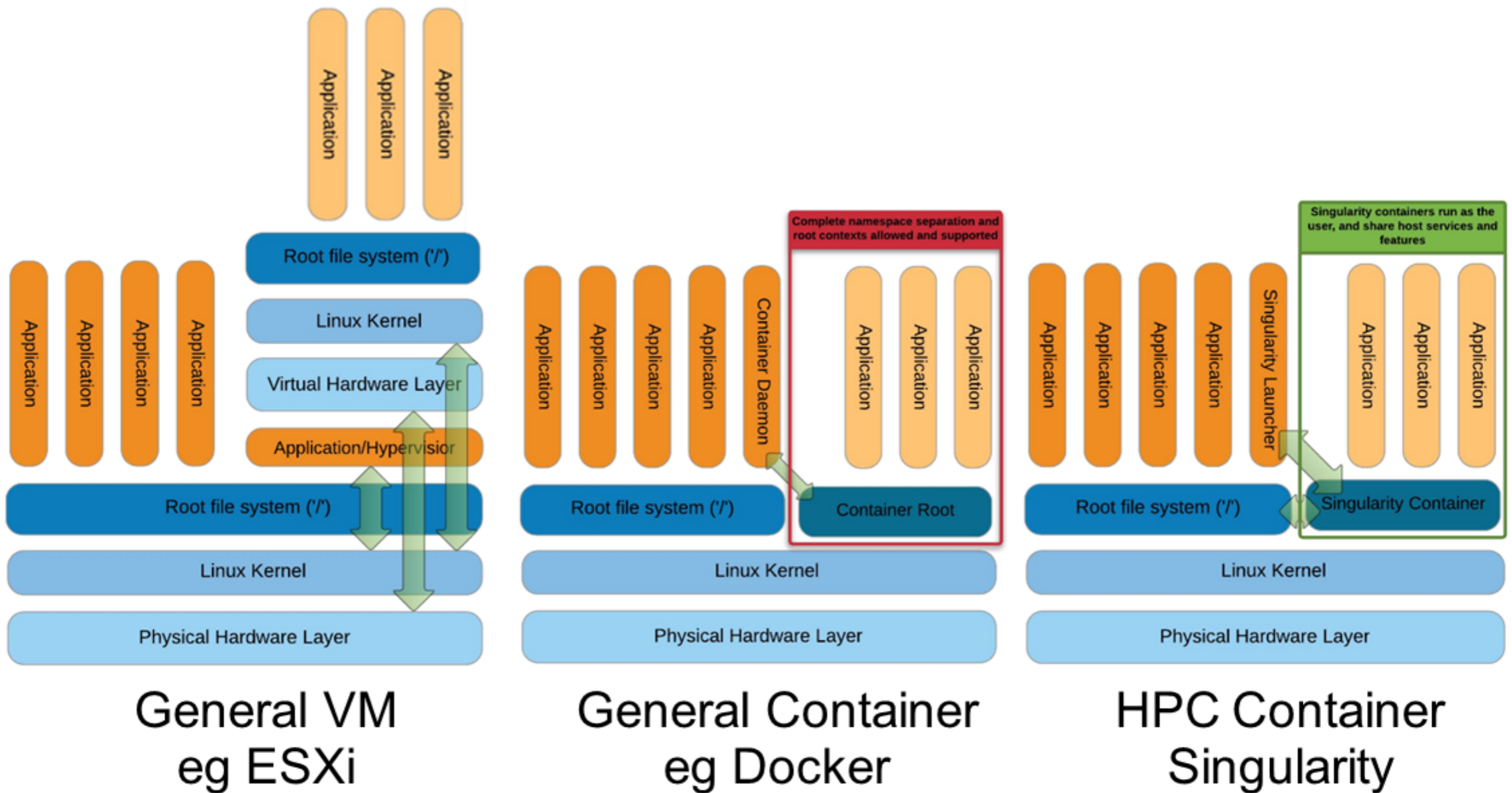
container

- Reuse the kernel
 - ➔ Software virtualisation



- ➔ Not multi os (linux only)
- ➔ fast/light
 - ➔ OK for single app
 - ➔ Good for HPC

HPC Container



Install Singularity

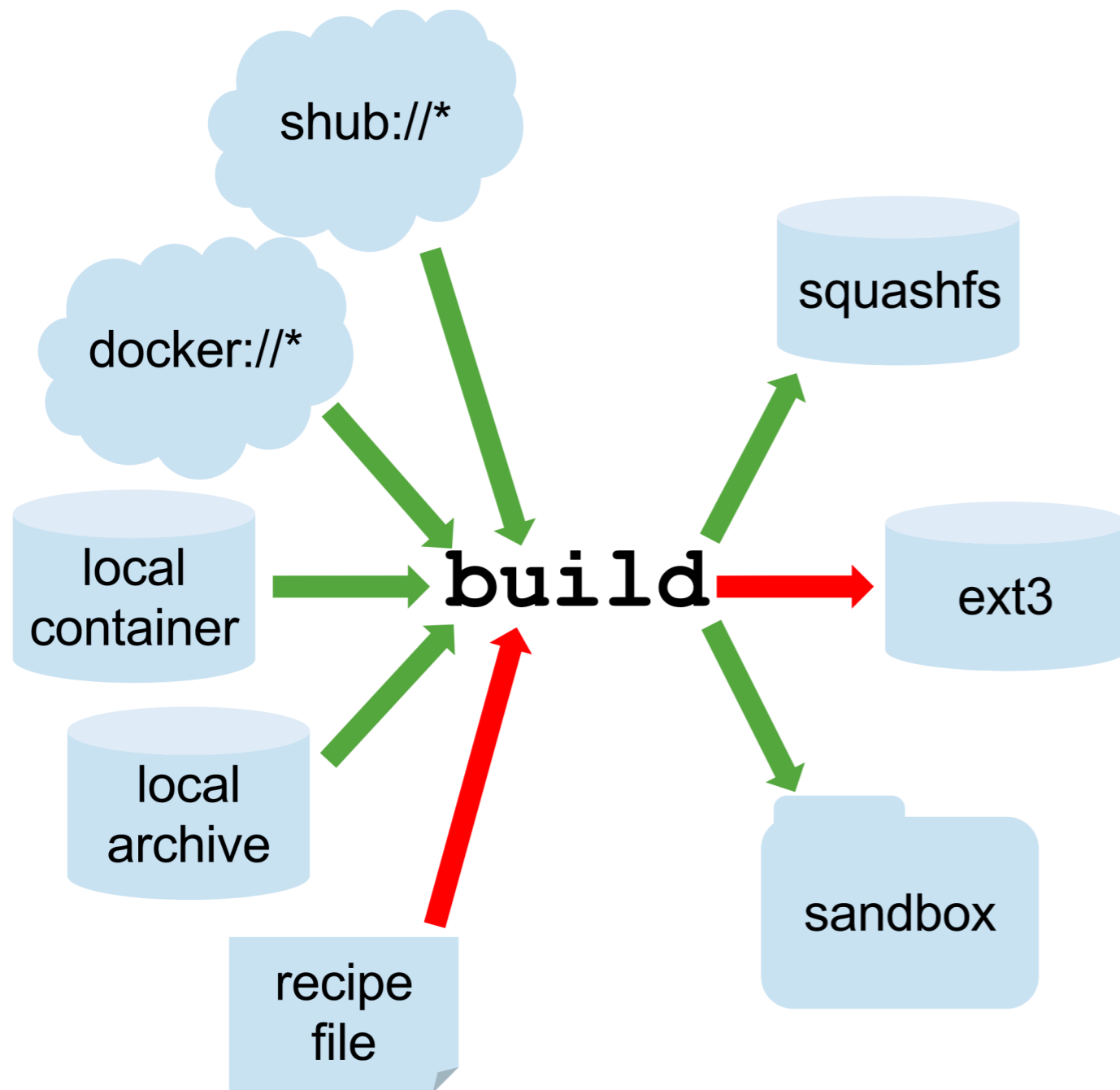
- This is linux only.
 - ➔ If you are on mac/windows you need to have a virtual machine
 - ◆ Easiest way install the one configure by singularity author
 - <https://singularity.lbl.gov/install-mac>
 - <https://singularity.lbl.gov/install-windows>
 - ◆ Or install Ubuntu on a VM and follow unix instruction
 - ➔ On linux:
 - ◆ <https://singularity.lbl.gov/install-linux>

Workflow

- Build
- Test
- Share
- Run

Building an image

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



- **squashfs:**
 - ➔ Read-only
 - ➔ Production
 - ➔ default
- **Ext3 --writable:**
 - ➔ Writable image
 - ➔ Single file
- **Sandbox --sandbox**
 - ➔ Full directory

Testing and Modifying image

```
vagrant@vagrant:~$ sudo singularity shell --writable hellocow/  
Singularity: Invoking an interactive shell within container...  
  
Singularity hellocow:~> █
```

- You can check that it has its own os:

```
Singularity hellocow:~> cat /etc/os-release  
NAME="Ubuntu"  
VERSION="16.04.3 LTS (Xenial Xerus)"  
ID=ubuntu  
ID_LIKE=debian  
PRETTY_NAME="Ubuntu 16.04.3 LTS"  
VERSION_ID="16.04"  
HOME_URL="http://www.ubuntu.com/"  
SUPPORT_URL="http://help.ubuntu.com/"  
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"  
VERSION_CODENAME=xenial  
UBUNTU_CODENAME=xenial  
Singularity hellocow:~> █
```

- If running without sudo
 - ➔ Can not become root in the image
- Running shell breaks reproducibility

Testing and Modifying image (II)

```
vagrant@vagrant:~/tuto2$ sudo singularity exec -w hello/ apt-get install emacs
```

- Test:

```
vagrant@vagrant:~/tuto2$ singularity shell hello/  
Singularity: Invoking an interactive shell within container...  
  
Singularity hello:~/tuto2> emacs  
Singularity hello:~/tuto2> which emacs  
/usr/bin/emacs  
Singularity hello:~/tuto2> |
```

- Allow to create an image step by step and keep a script with all modification
 - ➔ What if we want something more powerful

Recipe file

```
BootStrap: debootstrap
OSVersion: stable
MirrorURL: http://ftp.us.debian.org/debian/

%runscript
    echo "This is what happens when you run the container..."

%post
    echo "Hello from inside the container"
    apt-get update
    apt-get -y install fortune cowsay lolcat
    apt-get clean

%environment
    export PATH=$PATH:/usr/games
    export LC_ALL=C
```

- Other keywords:

- ➔ files

- ➔ setup (code running on the host)

- ➔ labels

- ➔ help

- ➔ test

- ➔ apps

Recover recipe file

```
vagrant@vagrant:~/tuto2$ singularity inspect --deffile hello/
BootStrap: debootstrap
OSVersion: stable
MirrorURL: http://ftp.us.debian.org/debian/

%runscript
    echo "This is what happens when you run the container..."

%post
    echo "Hello from inside the container"
    apt-get update
    apt-get -y install fortune cowsay lolcat
    #          apt-get clean

%environment
    export PATH=$PATH:/usr/games
    export LC_ALL=C

vagrant@vagrant:~/tuto2$
```

Update image with recipe/inspect

- Inspect command can also show you label/info

➔ Let's update our recipe with some label

```
%labels
```

```
author Olivier Mattelaer
```

- We do not have to re-run the full build!

```
vagrant@vagrant:~/tuto2$ sudo singularity build --section labels -w hello Singularity
Building into existing container: hello
```

```
vagrant@vagrant:~/tuto2$ singularity inspect hello
{
  "org.label-schema.usage.singularity.deffile.bootstrap": "debootstrap",
  "org.label-schema.usage.singularity.deffile": "Singularity",
  "AUTHOR": "Olivier Mattelaer",
  "org.label-schema.schema-version": "1.0",
  "org.label-schema.usage.singularity.deffile.osversion": "stable",
  "org.label-schema.build-date": "Tue,_30_Oct_2018_09:42:27_+0000",
  "org.label-schema.usage.singularity.deffile.mirrorurl": "http://ftp.us.debian.org/debian/",
  "org.label-schema.usage.singularity.version": "2.5.2-dist",
  "org.label-schema.build-size": "829MB"
}
```

- You can check the --deffile has been updated

Run with image

```
vagrant@vagrant:~/tuto2$ singularity exec hello cowsay 'I am a cow'
-----
< I am a cow >
-----
      \   ^__^
       (oo)\_____
          (__)\       )\/\
              ||----w |
              ||     ||
```

- Shell/piping works as normal

```
vagrant@vagrant:~/tuto2$ singularity exec hello cowsay 'I am a cow' > content
vagrant@vagrant:~/tuto2$ ls
content  GodLoveD-lolcow-master-latest.simg  hello  output  Singularity  Singularity~
vagrant@vagrant:~/tuto2$ cat content
-----
< I am a cow >
-----
      \   ^__^
       (oo)\_____
          (__)\       )\/\
              ||----w |
              ||     ||
```

- As said before filesystem is the one of the host

```
vagrant@vagrant:~/tuto2$ singularity exec hello /bin/touch cowsay_now
vagrant@vagrant:~/tuto2$ ls
content  cowsay_now  GodLoveD-lolcow-master-latest.simg  hello  output  Singularity  Singularity~
vagrant@vagrant:~/tuto2$
```

Run with image

- Image are executable! (not --sandbox)
 - ➔ ./lolcow.simg
 - ➔ Run the “%runscript” part of the definition file!
 - ◆ Behave as an app
 - Think of putting help/...

```
%runscript
    python /usr/local/bin/helloworld.py $@

%post
    echo "Hello from inside the container"
    apt-get update
    apt-get -y install python
    #          apt-get clean

%files
    helloworld.py /usr/local/bin
```

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
 - ➔ This file is a local directory of my MAC!
- Also possible via environment variable:
 - ➔ `export SINGULARITY_BINDPATH=/vagrant:/mnt`

Share

- You can share your image via singularity hub.
 - ➔ You can even build them on that platform
- Create a git repo
- Go to singularity hub -> connect to your git account
- Link your project and include recipe (Singularity.tag)
 - ➔ Wait they create the image

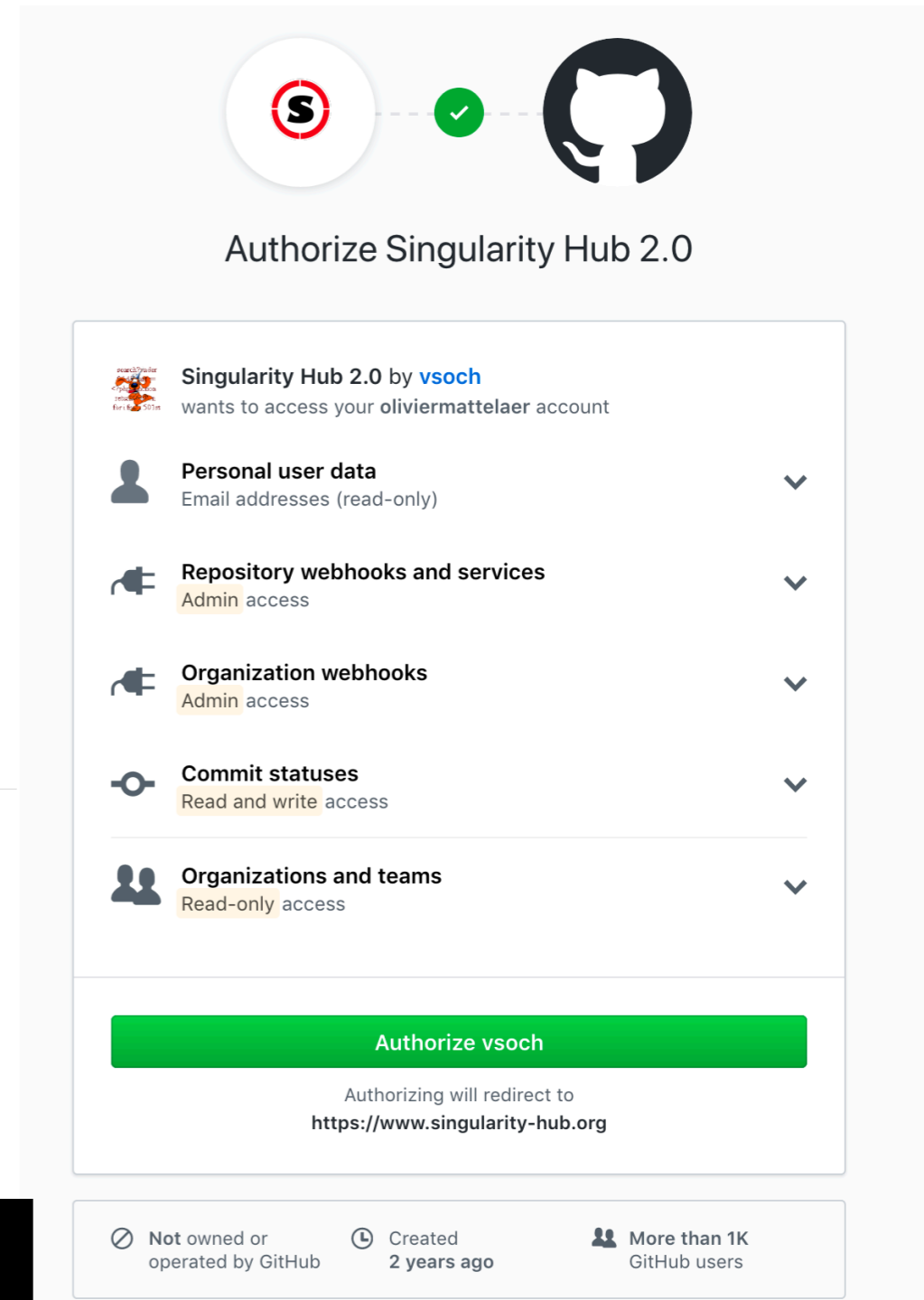
Builds COMMIT

uri ↓	Recipe	Status	Tag (Branch)	Date
  oliviermattelaer/singularity-recipe:singul...	python.Singularity 	RUNNING	singularity (master)	Oct. 30, 2018, 7:56 a.m. commit

- You can share it:
 - Singularity run shub://owner/collection:tag

```
vagrant@vagrant:~/tutorial$ singularity run shub://oliviermattelaer/singularity-recipe:cowsay
Progress |=====| 100.0%
This is what happens when you run the container...
Usage:
oliviermattelaer-singularity-recipe-master-cowsay.simg -i <infile> -o <outfile> [ -- <cowsay options> ]
vagrant@vagrant:~/tutorial$
```

- Singularity pull to download an image (no need of sudo)



Authorize Singularity Hub 2.0

Singularity Hub 2.0 by vsoch wants to access your oliviermattelaer account

- Personal user data
Email addresses (read-only)
- Repository webhooks and services
Admin access
- Organization webhooks
Admin access
- Commit statuses
Read and write access
- Organizations and teams
Read-only access

[Authorize vsoch](#)

Authorizing will redirect to <https://www.singularity-hub.org>

Not owned or operated by GitHub | Created 2 years ago | More than 1K GitHub users

MPI

- The network is shared between the host and the container
 - ➔ In principle MPI should not be an issue for container
 - ➔ In practise, you have to have the same library within the container / cluster **breaking the portability.**

Hands-on Session

- Follow the tutorial at the following page:
 - ➔ <https://github.com/oliviermattelaer/Singularity-Tutorial>
- Goodies to WIN !!!
 - ➔ Ask question to win one.

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...
 - ◆ We plan to offer (virtual) machine dedicated to the image building where you will be allowed to run
 - ◆ Sudo singularity