

# Introduction to Singularity containers

ABACBS 2018 Container Workshop

James Breen & Jan P. Buchmann

Jan P. Buchmann

[jan.buchmann@sydney.edu.au](mailto:jan.buchmann@sydney.edu.au)

The University Of Sydney

2018-11-29

# Just in case...

Download

<https://www.sylabs.io/singularity/get-singularity/>

Manual

<https://www.sylabs.io/guides/3.0/user-guide.pdf>

Paper

<https://doi.org/10.1371/journal.pone.0177459>

# Singularity: Containers for HPC

Containers are encapsulated system environments

**Not a microservice:** Scientific focus, e.g. whole pipelines

**Single file:** The image is a single file easily share, archive, reproduce, good for parallel file systems, e.g. Lustre

**Run as user:** root to create, user to run

**Access HPC resources:** MPI, GPUs, InfiniBand/Network, file systems

# Biggest difference to Docker

## Privileges

You run the container as the user who invokes singularity. You can only be root in the container if you run it as root. Not your usual HPC experience.

## No daemon

There is no daemon required, Singularity image is mounted as a loopback. Docker swarms need a DockerEngine on each node or instance they run.

## Runs closer to the host

Running a singularity container bind mounts your \$HOME, /dev, /sys, /tmp, /var/tmp, and /proc by default.

## Expect ongoing changes

Expect ongoing changes, e.g. differences singularity 2.2.5 to 3.0.1. Keeps you on your toes.

## Docker layers change

Docker image layers can change, i.e. a docker image has likely changed layers when pulled a couple of months later.

# Speed

Negligible tradeoffs compared to bare metal.

▶ <https://arxiv.org/pdf/1709.10140.pdf>

# Overall Singularity workflow

1. Build basic image
2. Modify basic image → your image
3. Copy your image to HPC, laptop, Grandma's fridge...
4. Run your image

# Building singularity containers

## Docker Hub (docker://)

```
singularity build lolcow.simg docker://godlovedc/lolcow
```

## Container Library (library://)

requires  $\geq 3.0$

```
singularity build lolcow.simg library://sylabs-jms/testing/lolcow
```

## Singularity Hub (shub://)

```
singularity build demo.simg shub://jasongallant/singularity_demosingularity
```

## Singularity recipe files

Roll your own

# Invoking singularity

## Run singularity

`singularity [global options] command [command options]`

For example:

`singularity -v build --sandbox /tmp/ubuntu docker://ubuntu:latest`

## Very useful commands

`$: singularity help`

For example:

`$: singularity help build`



# Workflow: Build environment (root)

## Interactive

```
1 sudo singularity build --sandbox /tmp/ubuntu \
2   docker://ubuntu:latest # In container
3 sudo singularity exec --writable /tmp/ubuntu/ \
4   apt-get update          # In container
5 sudo singularity exec --writable /tmp/ubuntu/ \
6   mkdir /mnt/builds       # In container
7 sudo singularity shell -B hostdir:/mnt/builds -w /tmp/ubuntu/
8 #-> compile, copy, etc. tools from hostdir in the container
9 sudo singularity build raxml.sif /tmp/ubuntu/
10 sudo singularity shell raxml.sif
```

# Workflow: Let's build ourself a container

## In your terminal

```
$: cd repo/singularity
```

```
$: mkdir work
```

```
$: cd work
```

Steps: ../commands/commands.md

## Workflow: Recipe file

- ▶ Preferred way to crate images. Use interactive to get details right, test dependencies, etc. than fixeverything in a receipe file.

Example: `../recipes/raxml.ubuntu.def`

### Run

recipes

```
$: cd repo/singularity/work
```

```
$: sudo singularity build ../recipes/raxml.ubuntu.def raxml.ubuntu.simg
```

# Singularity on HPC

- ▶ Host and container OS need same OpenMPI/MPI version
- ▶ Adjust image for binding directories
- ▶ Adjust GPU specific files, libraries
- ▶ Very useful: <https://hpc.nih.gov/apps/singularity.html>

Example PBS file: `../recipes/raxml-sing.pbs`

## Possible error

### FATAL: kernel too old

The glibc version in the image is too new for the host kernel.  
Update host kernel (not always possible) or use image with older glibc.

### FATAL: Unable to create build: unable to get conveyorpacker: invalid build source

Can occur with newer version (4537182). Bootstrap needs to be first line. Remove all comments until Bootstrap:...  
(../examples/{raxml.ubuntu.def,raxml.ubuntu.git.def})