Singularity

Michael Bauer

Contact Information

Michael Bauer

Staff Engineer, RStor
michael.bauer@rstor.io

@bauerm97 on GitHub
bauerm@umich.edu

Singularity

Singularity enables users to have full control of their environment. This means that a non-privileged user can “swap out” the operating system on the host for one they control. So if the host system is running RHEL6 but your application runs in Ubuntu, you can create an Ubuntu image, install your applications into that image, copy the image to another host, and run your application on that host in it’s native Ubuntu environment!

Singularity

Information
Download / Installation
Contributing
Getting Help
Documentation

Register your Cluster Add a Publication

Singularity also allows you to leverage the resources of whatever host you are on. This includes HPC interconnects, resource managers, file systems, GPUs and/or accelerators, etc. Singularity does this by enabling several key facets:

- Encapsulation of the environment
- Containers are image based
- No user contextual changes or root escalation allowed
- No root owned daemon processes

Getting started

Jump in and **get started**.

What are Containers?

Containers

… are encapsulations of system environments (software, libraries, etc…)

… allow portability of workflows between resources

… are lightweight and introduce little overhead

Containers


ubuntu
POWERED BY
APACHE
NGINX
http server
PostgreSQL
MariaDB
CentOS

Containers for Scientific Computing

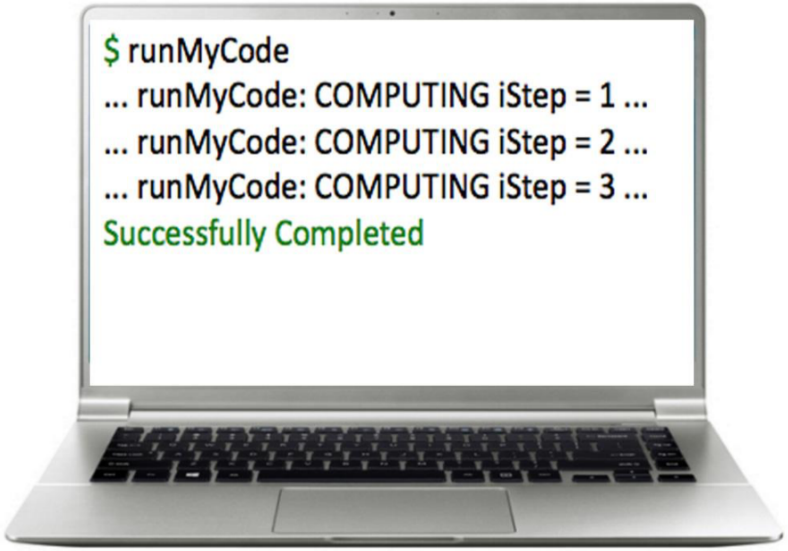
Why do we want containers in HPC?

Escape “dependency hell”

Local and remote code works identically every time

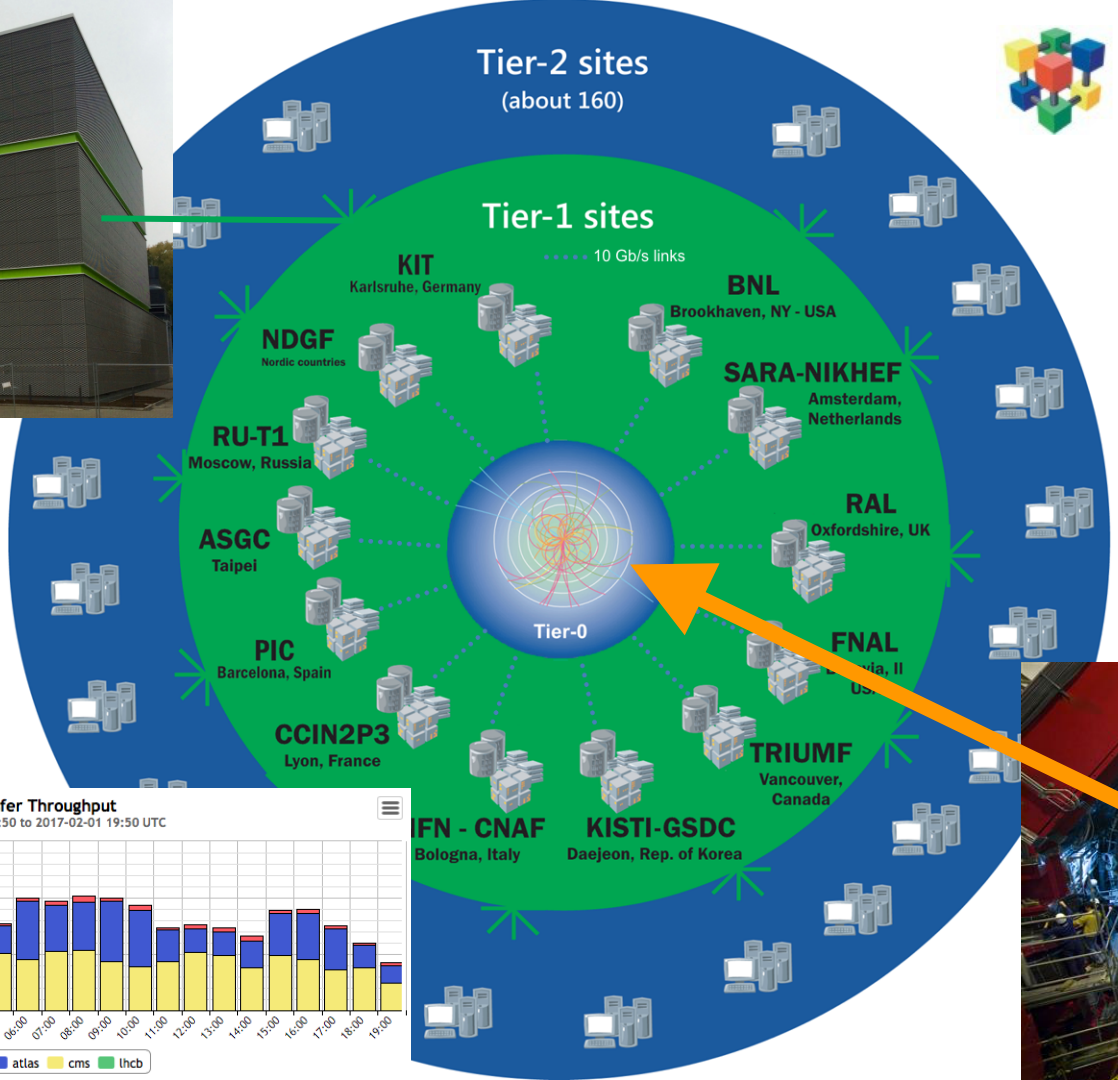
One file contains everything and can be moved anywhere

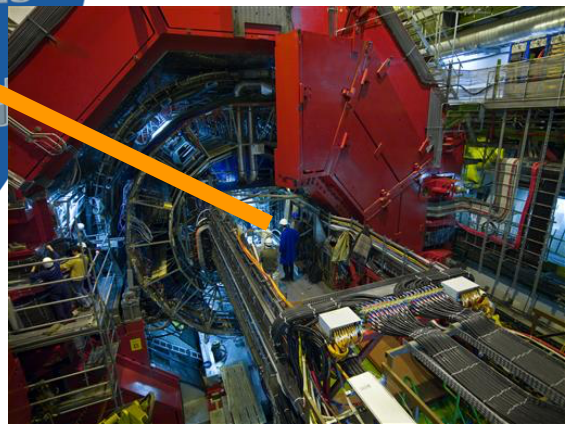
Environment Matters



ALICE Tier 2 Use Case

GSI Green Cube
Darmstadt
Germany





ALICE Detector LHC
Geneva
Switzerland

ALICE Tier 2: Problem

Run ALICE jobs on ~2k jobs at any time

Host machines run Debian 7.x kernel 3.16

ALICE expects Scientific Linux 6 (SL6)

Library incompatibilities cause frequent errors (much higher than expected)

ALICE Tier 2: Current Solution

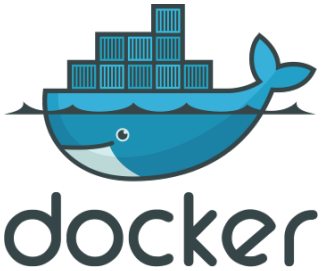
Correct library versions mounted in Lustre

SLURM job submission script alters $LD_LIBRARY_PATH to point to Lustre

Big Ugly Hack

Docker?

Docker



… is the most well known and utilized container platform

… is designed primarily for network micro-service virtualization

… facilitates creating, maintaining and distributing container images

… containers are kinda reproducible

… is easy to install, well documented, standardized

But I want to keep using Docker!

The good news:

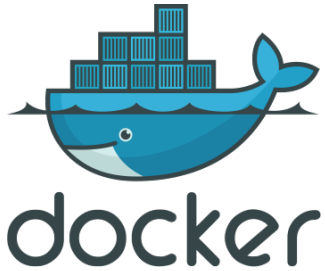
You can! It works great for local and private resources. You can use it to develop and share your work with others using Docker-hub.

The bad news:

If you ever need to scale beyond your local resources, it maybe a dead end path! Docker, and other enterprise focused containers, are not designed for, efficient or even compatible with traditional HPC.

No HPC centers allow it!

Needs for HPC containers



Any user can run containers without special privileges (root)

Integrate seamlessly into existing infrastructure

Portability between many systems

Users created and provided containers (no administrative oversight)

HPC container software can never touch root


NOT
A
THRU
ROAD
YOUR GPS
IS WRONG!
DEAD END

Singularity

Needs for HPC containers

- Any user can run containers without special privileges (root) ✓
- Integrate seamlessly into existing infrastructure ✓
- Portability between many systems ✓
- Users created and provided containers (no administrative oversight) ✓

Singularity

Any container can be run by any user - same user inside container and on host

No workflow changes necessary to use

Single .img file contains everything necessary

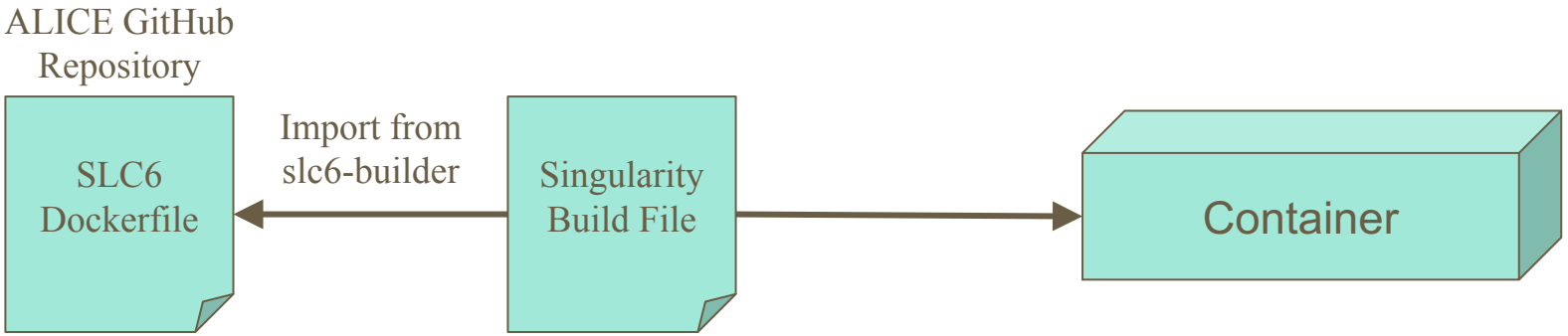
Safe to run any container without screening its contents

ALICE Tier 2: Singularity Solution

Package Scientific Linux 6 into container

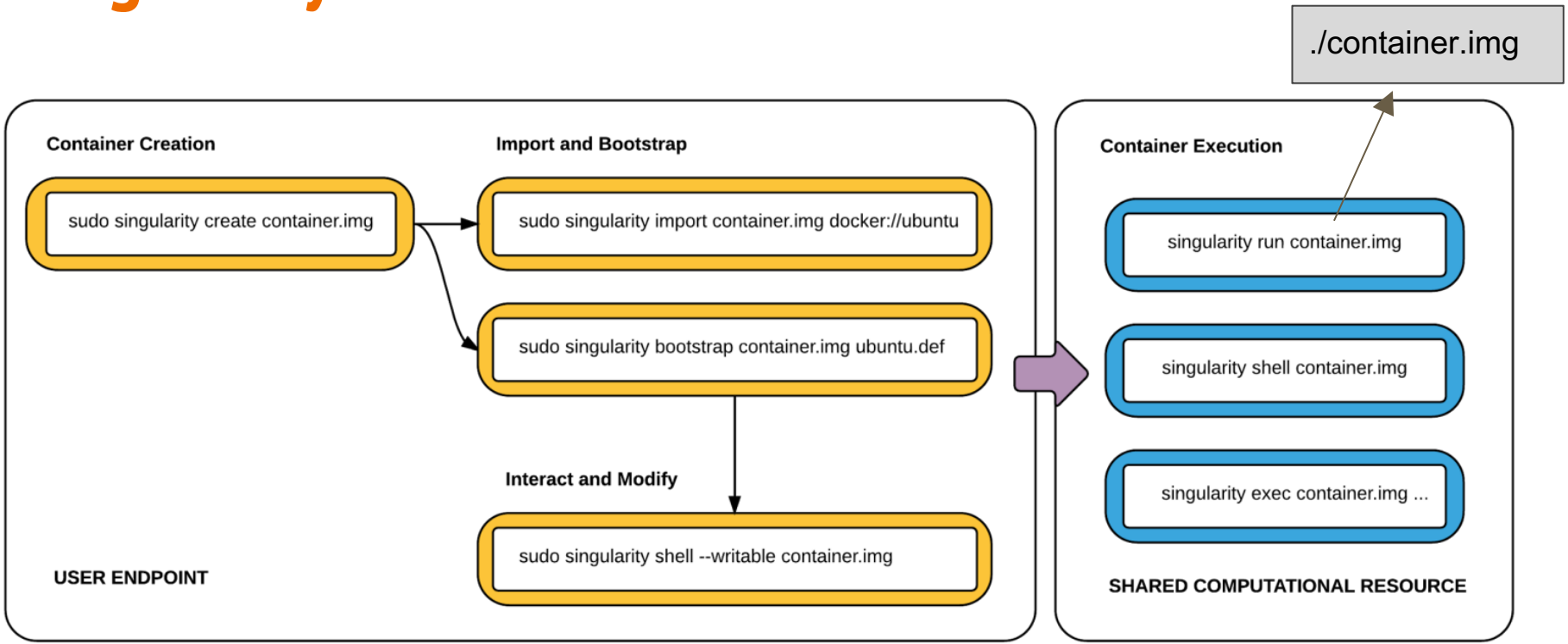
Modify SLURM submission script to run container

Can test container locally before deploying to HPC



Basic Usage of Singularity

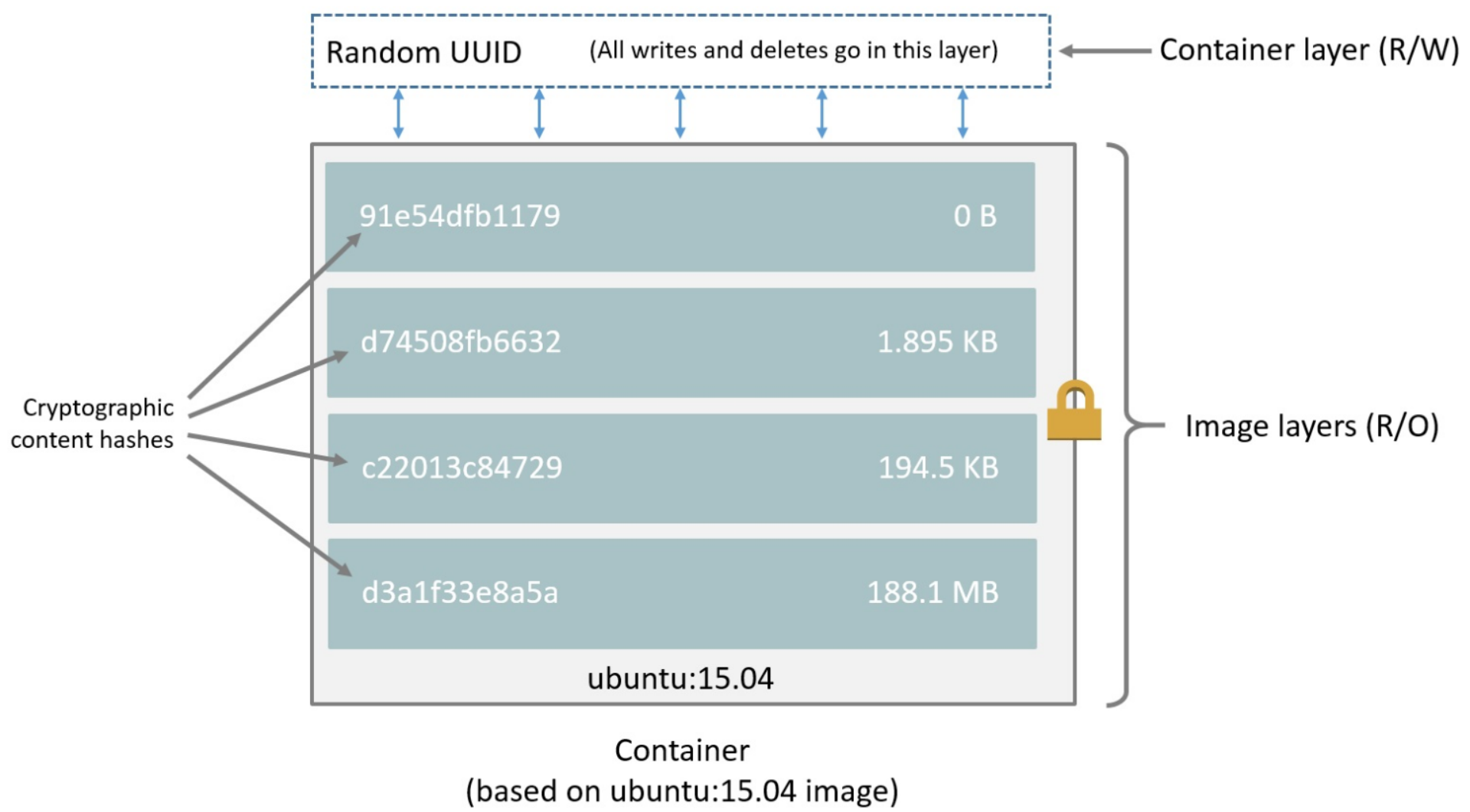
Singularity: Workflow



Kurtzer GM, Sochat V, Bauer MW. Singularity: Scientific Containers for Mobility of Compute

Format	**Description**
directory	Standard Unix directories containing a root container image
tar.gz	Zlib compressed tar archives
tar.bz2	Bzip2 compressed tar archives
tar	Uncompressed tar archives
cpio.gz	Zlib compressed CPIO archives
cpio	Uncompressed CPIO archives

Docker Integration in Singularity


Random UUID
(All writes and deletes go in this layer)
Container layer (R/W)
91e54dfb1179
0 B
d74508fb6632
1.895 KB
c22013c84729
194.5 KB
d3a1f33e8a5a
188.1 MB
ubuntu:15.04
Cryptographic content hashes
Image layers (R/O)
Container
(based on ubuntu:15.04 image)

```
$ singularity exec docker://python:latest /usr/local/bin/python hello.py
library/python:latest
Downloading layer: sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4
Downloading layer: sha256:e41da2f0bac3da1769ecdac8b0f5df53c1db38603e39b9e261cafd10caf904de
Downloading layer: sha256:75ef15b2048b4cfb06c02f2180f4d89033d02c63f698672d2909b8c9878c4270
Downloading layer: sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4
Downloading layer: sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4
Downloading layer: sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4
Downloading layer: sha256:45b2a7e03e44b5ea7fad081537134c9cc725bddf94f9093b00e1fa8d8ebbcda1
Downloading layer: sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4
Downloading layer: sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4
Downloading layer: sha256:52f3db4b5710849a53bc2eea0b6f0895c494d751c38c597404d805da82b3f37c
Downloading layer: sha256:76610ec20bf5892e24cebd4153c7668284aa1d1151b7c3b0c7d50c579aa5ce75
Downloading layer: sha256:fce5728aad85a763fe3c419db16885eb6f7a670a42824ea618414b8fb309ccde
Downloading layer: sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4
Downloading layer: sha256:5040bd2983909aa8896b9932438c3f1479d25ae837a5f6220242a264d0221f2d
Hello World: The Python version is 3.6.0
```

```
$ singularity exec docker://tensorflow/tensorflow python -m tensorflow.models.image.mnist.convolutional
tensorflow/tensorflow:latest
Downloading layer: sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4
...
Downloading layer: sha256:6498e51874bfd453352b79b1a3f669109795134b7adcd1a02d0ce69001f4e05b
Downloading layer: sha256:862a3e9af0aeffe79345b790bad31baaa61e9402b6e616bff17babed6b053b54
Successfully downloaded train-images-idx3-ubyte.gz 9912422 bytes.
Successfully downloaded train-labels-idx1-ubyte.gz 28881 bytes.
Successfully downloaded t10k-images-idx3-ubyte.gz 1648877 bytes.
Successfully downloaded t10k-labels-idx1-ubyte.gz 4542 bytes.
Extracting data/train-images-idx3-ubyte.gz
Extracting data/train-labels-idx1-ubyte.gz
Extracting data/t10k-images-idx3-ubyte.gz
Extracting data/t10k-labels-idx1-ubyte.gz
Initialized!
Step 0 (epoch 0.00), 5.1 ms
Minibatch loss: 8.334, learning rate: 0.010000
Minibatch error: 85.9%
Validation error: 84.6%
Step 100 (epoch 0.12), 140.0 ms
Minibatch loss: 3.250, learning rate: 0.010000
Minibatch error: 6.2%
Validation error: 7.6%
…
Step 8500 (epoch 9.89), 134.2 ms
Minibatch loss: 1.618, learning rate: 0.006302
Minibatch error: 0.0%
Validation error: 0.9%
Test error: 0.8%
```

SLURM Integration

```
#!/bin/bash -l

#SBATCH --image=~/centos7/latest
#SBATCH -p debug
#SBATCH -N 64
#SBATCH -t 00:20:00
#SBATCH -J my_job
#SBATCH -L SCRATCH
#SBATCH -C haswell

srun -n 4096 ./mycode.exe       # an extra -c 1 flag is optional for fully packed pure MPI with hyperthreading
```

Thank you! Questions?

Global Options	
−d − −debug	Print debugging information
−h − −help	Display usage summary
−q − −quiet	Only print errors
− − version	Show application version
−v − −verbose	Increase verbosity +1
−x − −sh − debug	Print shell wrapper debugging information
General Commands	
help	Show additional help for a command
Container Usage Commands	
exec	Execute a command within container
run	Launch a runscript within container
shell	Run a Bourne shell within container
test	Execute any test code defined within container
Container Management Commands (requires root)	
bootstrap	Bootstrap a new Singularity image
copy	Copy files from your host into the container
create	Create a new container image
export	Export the contents of a container via a tar pipe
import	Import/add container contents via a tar pipe
mount	Mount a Singularity container image