Singularity

Portability of containers across diverse HPC resources with Singularity

Michael Bauer
César Gómez-Martín

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!

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.

Singularity

Information

Download / Installation

Contributing

Getting Help

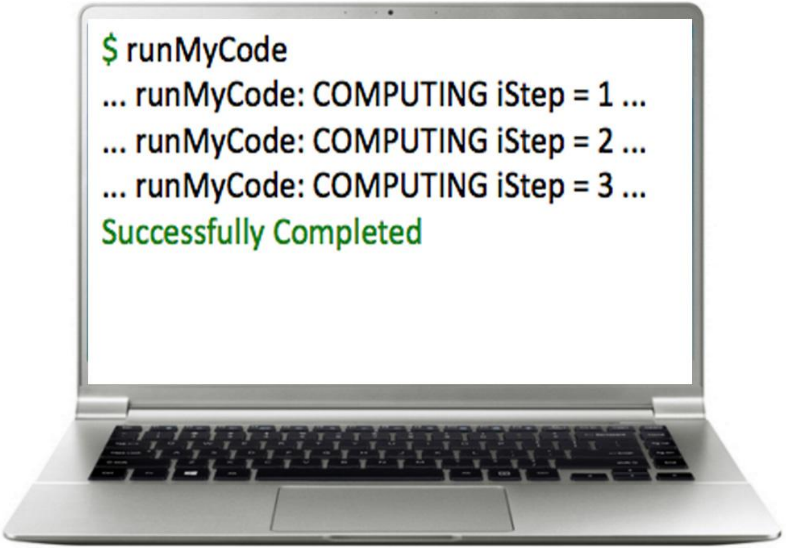
Documentation

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



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

Site or Organization	System Name	Size (cores)	Purpose of the System
CSIRO	bragg-gpu	2048	broad base scientific
GSI Helmholtz Center	Greencube	300,000	Heavy Ion Physics
Holland Computing Center at UNL	Crane and Tusker	14,000	General purpose campus cluster
HPC-UGent	golett	2500	research across all scientific domains
Lunarc	Aurora	360	Research
Microway	Microway Research Cluster	192	Scientific benchmarking
MIT	openmind	1,176	Neuroscience
National Institute of Health HPC	Biowulf	54,000	General purpose biomedical research
Purdue University	Rice	11520	Campus HPC resource
Purdue University	Conte	78880	Campus HPC resource
Purdue University	Snyder	2220	Campus HPC resource
Purdue University	Hammer	3960	Campus HPC resource
Purdue University	Carter	10560	Campus HPC resource
R Systems NA, Inc.	Oak1	1024	Shared commercial/academic resource
R Systems NA, Inc.	Oak2	2048	Shared commercial/academic resource
R Systems NA, Inc.	HOU1	5376	Shared commercial/academic resource
San Diego Supercomputer Center	Gordon	16384	HPC cluster for XSEDE users
San Diego Supercomputer Center (SDSC)	Comet	47776	HPC Cluster for XSEDE users
Texas Advanced Computing Center	Stampede	102400	NSF key resource, all fields
UFIT Research Computing at the UF	HiPerGator	51,000	research computing cluster
Ulm University, Germany	JUSTUS	550	Computational Chemistry
University of Chicago	midway.rcc.uchicago.edu	24196	University cluster
University of Manitoba	Grex	3840	General purpose HPC cluster
Georgia State University	Orion	362	research
UNF	Stark	64	Functional MRI analysis of the Brain
Genentech, Inc.			Research
Rutgers University	sirius	32	scientific SMP machine
Stanford University	sherlock	12764	Compute for Stanford researchers
Stanford University	scg4	3920	Genomics at Stanford
The University of Leeds	MARC1	1236	Bioinformatics, data analytics
McGill HPC Centre/Calcul Québec	guillimin	22300	Compute Canada cluster
University of Arizona	Ocelote	10000	General Research
University of Arizona	ElGato	2300	GPU cluster
Washington University in St. Louis		2000	General purpose cluster

Basic Usage of Singularity

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

Singularity Workflow

1. Create image file
   - $ sudo singularity create [image]
2. Bootstrap image
   - $ sudo singularity bootstrap [image] [definition.def]
3. Run image
   - $ singularity shell [image]
   - $ singularity exec [image] [/path/to/executable]
   - $ singularity run [image]
   - $ ./image

	Shifter	Charlie Cloud	Docker	Singularity
Privilege model	SUID	UserNS	Root Daemon	SUID/UserNS
Support current production Linux distros	Yes	No	No	Yes
Internal image build/boostrap	No*	No*	No**	Yes
No privileged or trusted daemons	Yes	Yes	No	Yes
No additional network configurations	Yes	Yes	No	Yes
No additional hardware	Maybe	Yes	Maybe	Yes
Access to host filesystem	Yes	Yes	Yes***	Yes
Native support for GPU	No	No	No	Yes
Native support for InfiniBand	Yes	Yes	No	Yes
Native support for MPI	Yes	Yes	No	Yes
Works with all schedulers	No	Yes	No	Yes
Designed for general scientific use cases	Yes	No	No	Yes
Contained environment has coorect perms	Yes	No	Yes	Yes
Containers are portable, unmodified by use	No	No	No	Yes
Trivial HPC install (one package, zero conf)	No	Yes	Yes	Yes
Admins can control and limit capabilities	Yes	No	No	Yes

* Relies on Docker

** Depends on upstream

*** With security implications

SHIFTER

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

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