Sylabs.io

Singularity Containers for Reproducible Workflows

David Trudgian Ph.D.

david.trudgian@sylabs.io

Sylabs.io

What is Singularity?

An open-source container runtime born out of the needs of HPC users

Developed by Greg Kurtzer at LBNL from 2015
Now led by Sylabs Inc. and the Singularity community

Allows you to build and run containers on an shared system, without needing to convince your administrator to install docker

Uses existing docker / OCI containers, or create your own Singularity containers
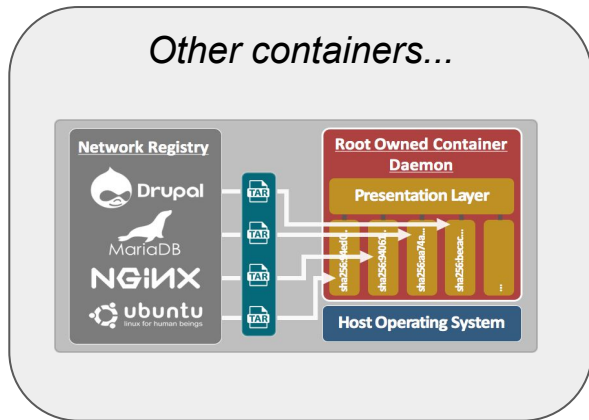
Sylabs.io

Designed Specifically for Compute

- Drop-in compatibility with nearly every environment

- Native support for: GPUs, Infiniband, MPI, parallel file systems, Cloud and Edge computing

- Optimized for performance and security

  - *https://insidehpc.com/2019/06/benchmarking-mpi-applications-in-singularity-containers-on-traditional-hpc-and-cloud-infrastructures/*

- Blocks privilege escalation within the container

  - Users run as themselves, not root

Sylabs.io

Singularity Image Format (SIF)

Other containers...

Singularity

Singularity Image Format (SIF) container provides:

➔ Single file (no intermediate layers to)
➔ Immutable base image
➔ Cryptographic signing & verification
➔ Reproducibility

Sylabs.io

Singularity Library

Share containers with others, or keep a private collection.

Tag containers or pull images by hash for reproducibility.

Free service currently offers 11GB quota.

Sylabs.io

Remote Builder

Build containers in the cloud, when you don't have a Linux machine handy, or can't use root / fakeroot.

Offers a way to ease builds for non-Intel systems. (On-premise enterprise version)

Free service offers 1,000 build minutes per month (rolling).

Sylabs.io

Keystore

Stores public keys against a Sylabs account.

Allow images pulled from the library to be verified as signed by an individual, and unmodified.

Free service

Get Singularity

Open Source: https://github.com/sylabs/singularity

For your Mac: https://sylabs.io/singularity-desktop-macos/

Paid Supported: https://sylabs.io/singularity/

(EPEL for RHEL / CentOS - community packaged, not from Sylabs)

Sylabs.io

Running a Container

```
02:20 PM $ singularity run library://godlovedc/funny/lolcow
 _____
/ Your talents will be recognized and \
\ suitably rewarded.                   /
 -------------------------------------
        \   ^__^
         \  (oo)_____
            (__)\       )\/\
                ||----w |
                ||     ||
```

```
02:29 PM $ singularity run docker://dctrud/wttr:latest
Weather report: Carrollton, United States

     \  /       Partly cloudy
   _ /"".-.     87..91 °F
     \_(   ).   ↑ 11 mph
     /(___(__)  9 mi
                0.0 in
```

			Thu 10 Oct		
Morning		Noon		Evenin	

\ /   Partly cloudy
_ /"".-.   73..77 °F
  _(   ).   ↑ 13-16 mph
  /(___(__) 6 mi
             0.0 in | 51%
		_`/"".-. Light rain sho…
		,_( ). 77..80 °F
		/(___(__) ↑ 14-18 mph
		           6 mi
		           0.0 in | 72%
				_`/"".-. Thu
				,_( ). 84.
				/(___(__) ↑ 1
				           5 m
				           0.0	

Sylabs.io

Accessing the Host's GPU With Tensorflow

```
$ singularity exec --nv docker://tensorflow/tensorflow:latest-gpu python
Python 2.7.12 (default, Dec  4 2017, 14:50:18)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> x1 = tf.constant([1,2,3,4])
>>> x2 = tf.constant([5,6,7,8])
>>> result = tf.multiply(x1, x2)
>>> print(result)
Tensor("Mul:0", shape=(4,), dtype=int32)
>>> exit()
$
```

Sylabs.io

Pulling a Container

```
02:32 PM $ singularity pull library://godlovedc/funny/lolcow
WARNING: Container might not be trusted; run 'singularity verify lolcow_latest.s
if' to show who signed it
INFO:    Download complete: lolcow_latest.sif
```

```
02:32 PM $ singularity verify lolcow_latest.sif
Container is signed by 1 key(s):

Verifying signature F: 4426F6CBCB38449E062940A42E6A34D6F54EBE59:
[REMOTE]   Dave Godlove (this is the best key) <d@sylabs.io>
[OK]       Data integrity verified

INFO:    Container verified: lolcow_latest.sif
```

Sylabs.io

Running from a SIF


```
02:32 PM $ singularity run lolcow_latest.sif

 _____
/ Your lucky number has been    \
\ disconnected.                  /
 -------------------------------
        \   ^__^
         \  (oo)_____
            (__)\       )\/\
                ||----w |
                ||      ||
```

Sylabs.io

Building from Docker to SIF

```
02:35 PM $ singularity build wttr.sif docker://dctrud/wttr:latest
INFO:    Starting build...
Getting image source signatures
Skipping fetch of repeat blob sha256:4a56a430b2bac33260d6449e162017e2b23076c6411
a17b46db67f5b84dde2bd
Skipping fetch of repeat blob sha256:82177f28f44660a30bd8a207c5b88b83c4be2329e71
```

```
23076c6411a17b46db67f5b84dde2bd
2019/10/10 14:35:42  info unpack layer: sha256:82177f28f44660a30bd8a207c5b88b83c
4be2329e71f92ad01f50febe1cf9caf
2019/10/10 14:35:42  info unpack layer: sha256:83a1cd2c9d1841681590aa66db329ade6
2a159dc0fa8f88534d8400861c6695d
INFO:    Creating SIF file...
INFO:    Build complete: wttr.sif
```

Building from Docker to SIF

Building from a Definition File

```
Bootstrap: docker

From: debian:buster


%post

    apt-get update

    apt-get -y install curl


%runscript

    curl -s https://wttr.in
```

Building from a Definition File

```
Bootstrap: docker          ⟵          Where to get our base container

From: debian:buster        ⟵          What base container to use


%post

    apt-get update                    Things to do during the build
                           ⟵
    apt-get -y install curl


%runscript
                                       What happens when we 'run'
    curl -s https://wttr.in ⟵
```

Sylabs.io

Building from a Definition File

```
03:07 PM $ sudo singularity build wttr_demo.sif wttr_demo.def
INFO:    Starting build...
Getting image source signatures
Copying blob sha256:4a56a430b2bac33260d6449e162017e2b23076c6411a17b46db67f5b84dde2bd
 48.05 MiB / 48.05 MiB [======================================================] 1s
Copying config sha256:9bd428a1eab6d94769f7e0d126c07a3b4170e53868e18a7c073ab785fb9b3a39
 579 B / 579 B [=============================================================] 0s
Writing manifest to image destination
Storing signatures
2019/10/10 15:07:58  info unpack layer: sha256:4a56a430b2bac33260d6449e162017e2b23076c6
INFO:    Running post scriptlet
+ apt-get update
Get:1 http://cdn-fastly.deb.debian.org/debian buster InRelease [122 kB]
Get:2 http://security-cdn.debian.org/debian-security buster/updates InRelease [39.1 kB]
```

--fakeroot   if you don't have sudo / root (requires user namespace)
--remote   if you need a remote build in the cloud

Sylabs.io

Sign….. and Verify

```
03:30 PM $ singularity sign wttr_demo.sif
Signing image: wttr_demo.sif
Enter key passphrase :
Signature created and applied to wttr_demo.sif
```

```
03:31 PM $ singularity verify wttr_demo.sif
Container is signed by 1 key(s):

Verifying signature F: 8C1158660B83F0AFEF7B1D5F91A6ECB189A8F00C:
[LOCAL]    David Trudgian (Piran Keys) <dave@trudgian.net>
[OK]       Data integrity verified


INFO:      Container verified: wttr_demo.sif
```
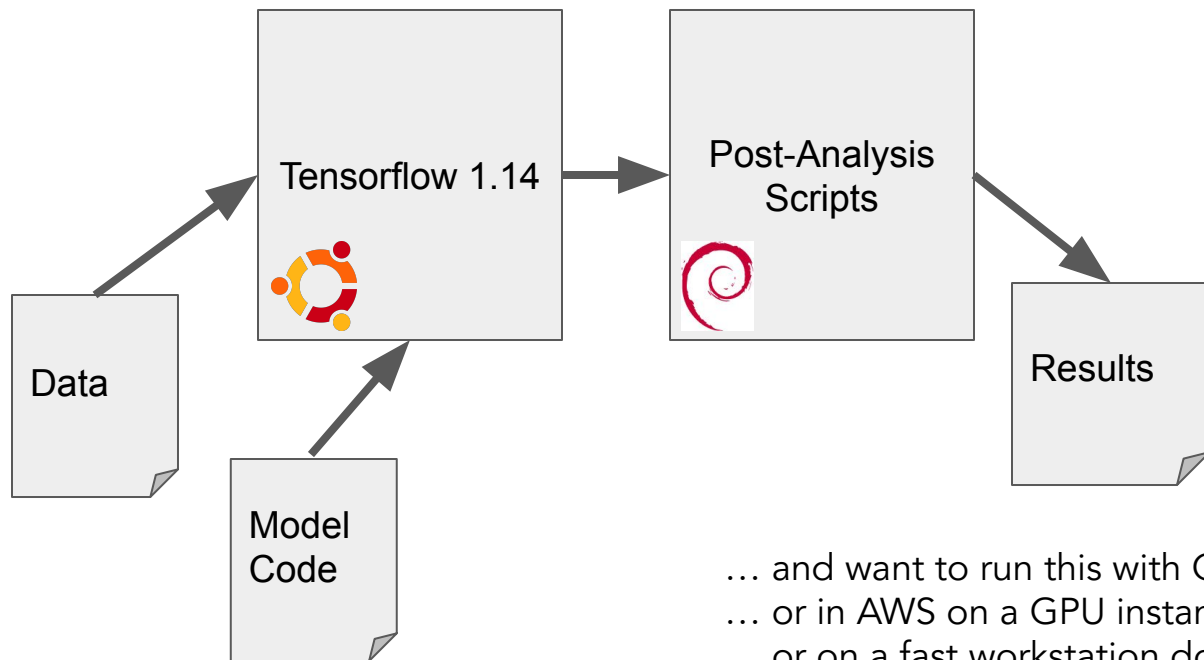
Sylabs.io

I need to run a tensorflow model, and some scripts I know work on my Debian VM…

… and want to run this with GPUs on a older CentOS 6 cluster
… or in AWS on a GPU instance
… or on a fast workstation down the hall

A Simple Workflow

Prepare…

```
singularity pull tflow114.sif docker://tensorflow/tensorflow:1.14.0-gpu

sudo singularity build myscripts.sif myscripts.def
```

Copy the containers and data to the cluster, then simply run / submit 2 lines:

```
singularity run --nv tflow114.sif mymodel.py mydata.in

singularity run myscripts.sif mymodel.out
```

Keep the containers and your data, and you can run it again, reproducibly & easily.

Sylabs.io

Native support for using Singularity containers in:
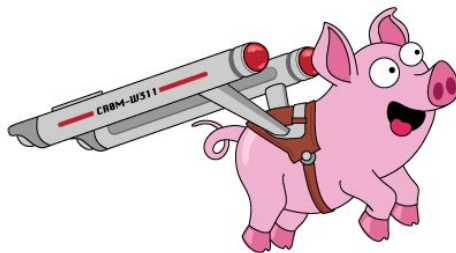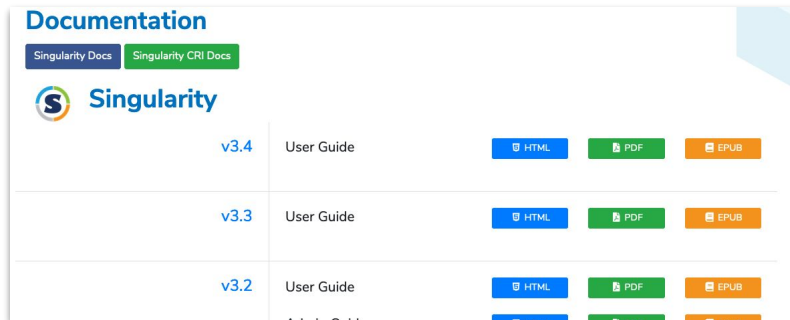
(cwltool, toil)

Documentation & Examples

https://sylabs.io/docs/

https://github.com/sylabs/examples

Sylabs.io

Community Resources

https://sylabs.io/resources/community

David Trudgian

david.trudgian@sylabs.io

Sylabs.io