Runtime

5th High Performance Container Workshop - ISC19

Scope and Introduction

This segment focuses on **RUNTIME** aspects

We do not talk about build and everything related to distribution.

The scope is a single nodes runtime.

5th Annual High Performance Container Workshop, ISC (June 20, 2019)

Innovative R&D by NTT

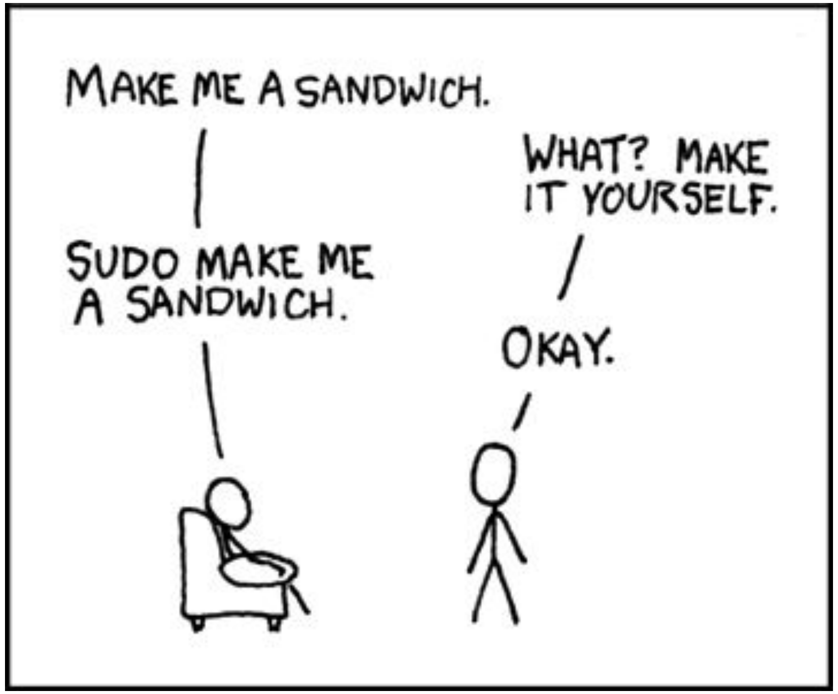
Current state of rootless dockerd

Akihiro Suda (@_AkihiroSuda_)
NTT Software Innovation Center

What is rootless dockerd?

- **Run Docker daemon (and also containers of course) as a non-root user**

- **Don't confuse with:**
  - `sudo`
  - `usermod -aG docker penguin`
  - `docker run --user`
  - `dockerd --userns-remap`



- **Experimentally supported since Docker v19.03**
  https://get.docker.com/rootless

Image: https://xkcd.com/149/

Why?

- **For Cloud-Native envs:**
  - To mitigate potential vulnerability of container runtimes and orchestrator

- **For HPC envs:**
  - To run containers without the risk of breaking other users environments

How it works: User Namespaces

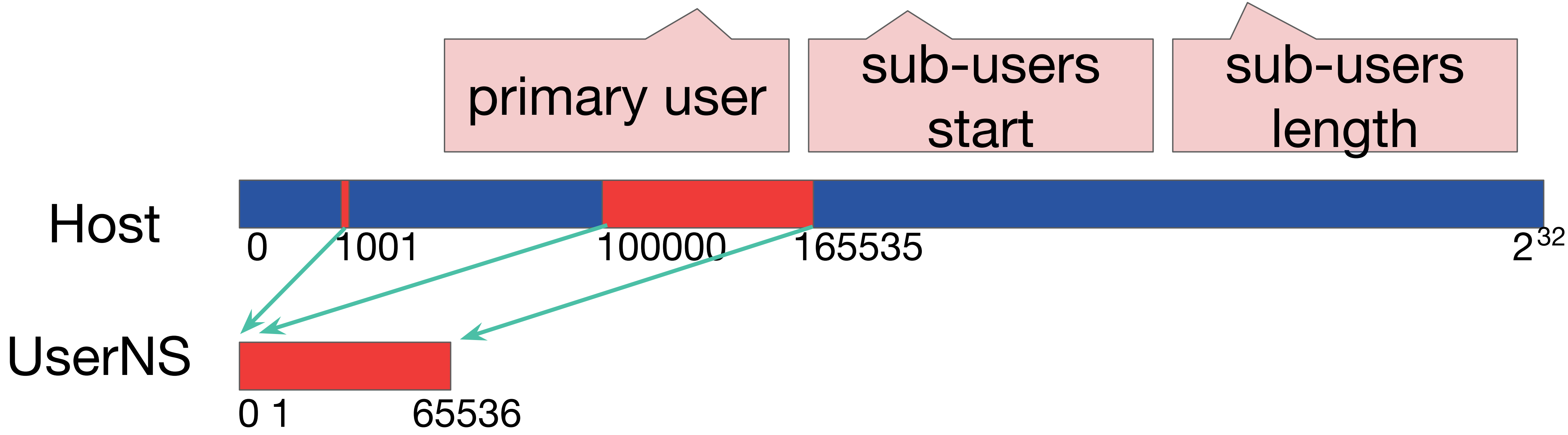
- **User namespaces allow non-root users to pretend to be the root**

- **Root-in-UserNS can have “fake” UID 0 and also create other namespaces (MountNS, NetNS..)**

- **Unlike Singularity, NetNS can be unshared**
  - By using either usermode TCP/IP stack (VPNKit, slirp4netns) or SETUID binary (lxc-user-nic)

System requirements: `/etc/{subuid,subgid}`

- **If `/etc/subuid` contains "`1001:100000:65536`"**

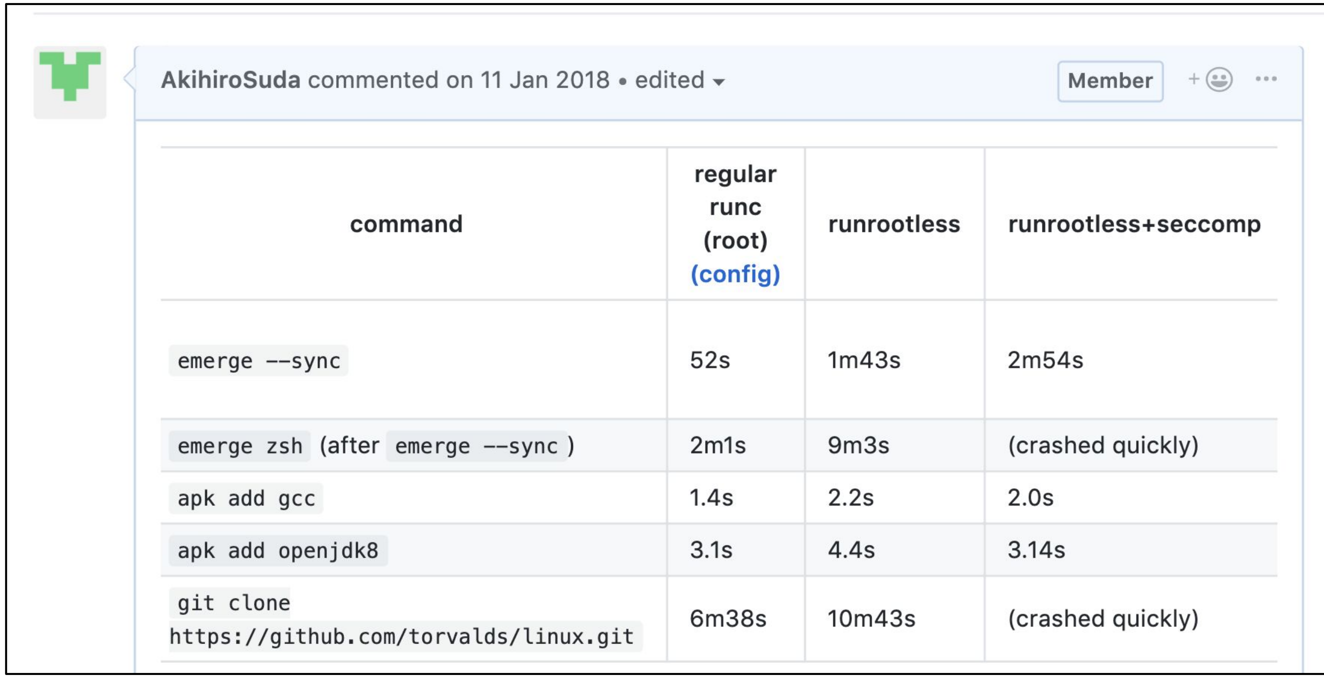


- **Having 65536 sub-users should be enough for most containers**

Unresolved issues (Contribution wanted!)

- **Hard to maintain `subuid` & `subgid` in LDAP/AD envs**
  - NSS module is being under discussion
    https://github.com/shadow-maint/shadow/issues/154
  - Single-mapping mode w/o `subuid` & `subgid` is also under discussion
    - uses ptrace and xattrs (slow!)
    - seccomp could be used for acceleration

    https://github.com/rootless-containers/runrootless



AkihiroSuda commented on 11 Jan 2018 • edited

command	regular runc (root) (config)	runrootless	runrootless+seccomp
`emerge --sync`	52s	1m43s	2m54s
`emerge zsh` (after `emerge --sync`)	2m1s	9m3s	(crashed quickly)
`apk add gcc`	1.4s	2.2s	2.0s
`apk add openjdk8`	3.1s	4.4s	3.14s
`git clone https://github.com/torvalds/linux.git`	6m38s	10m43s	(crashed quickly)

Unresolved issues (Contribution wanted!)

- **Lacks cgroup**
  - cgroup2 (unified-mode) supports unprivileged mode but migration may take a few years… or even more
  - For cgroup1, `pam_cgfs` could be used instead, but not available in Fedora / RHEL due to a security concern

- **Kernel / VM / HW may have vulns**
  - Not suitable for real multi-tenancy
  - gVisor might able to mitigate some of them

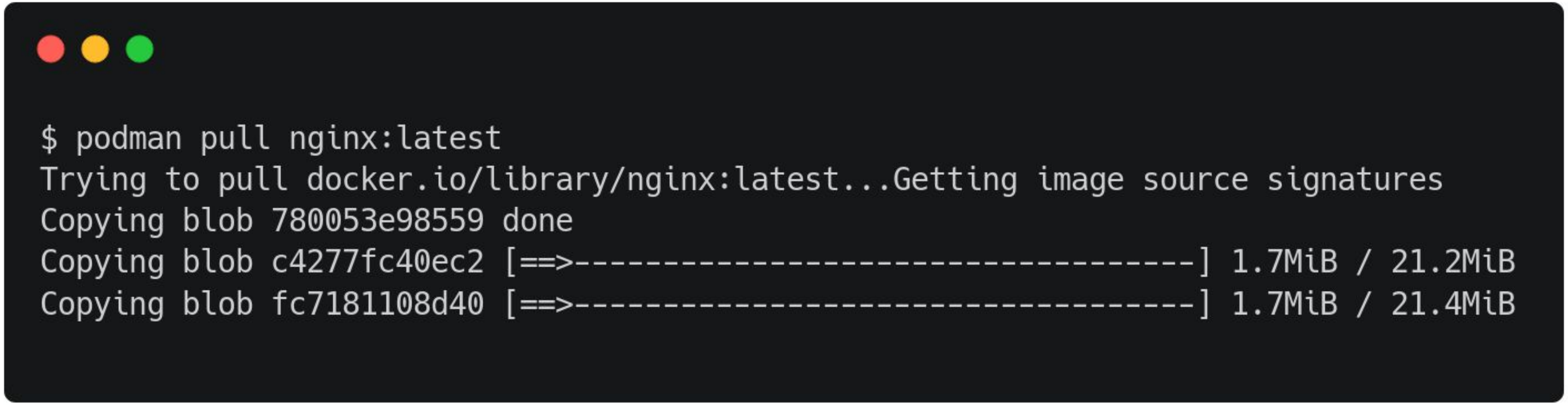
podman

Valentin Rothberg <rothberg@redhat.com>
@vlntnrthbrg

What is Podman?

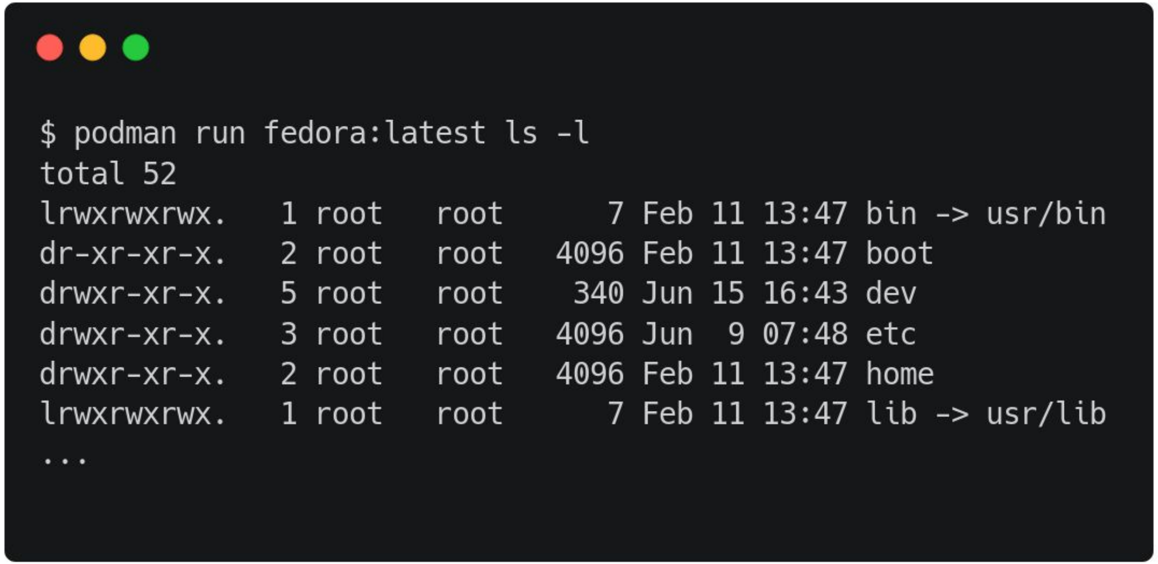
- Podman is a tool for managing pods and containers
- The CLI is based on Docker
  - Defacto standard CLI for managing containers
  - Allows for an easier transition of users *and* tools
- Developed at *github.com/containers/libpod*
  - *github.com/containers/image* for image management
  - *github.com/containers/storage* for local storage (overlay, btrfs, vfs, etc.)
  - *github.com/containers/buildah* for building images (O)

Podman - optimized image pulling



```
$ podman pull nginx:latest
Trying to pull docker.io/library/nginx:latest...Getting image source signatures
Copying blob 780053e98559 done
Copying blob c4277fc40ec2 [==>-----------------------------------] 1.7MiB / 21.2MiB
Copying blob fc7181108d40 [==>-----------------------------------] 1.7MiB / 21.4MiB
```

Podman - CLI compatibility



```
$ podman run fedora:latest ls -l
total 52
lrwxrwxrwx.   1 root   root      7 Feb 11 13:47 bin -> usr/bin
dr-xr-xr-x.   2 root   root   4096 Feb 11 13:47 boot
drwxr-xr-x.   5 root   root    340 Jun 15 16:43 dev
drwxr-xr-x.   3 root   root   4096 Jun  9 07:48 etc
drwxr-xr-x.   2 root   root   4096 Feb 11 13:47 home
lrwxrwxrwx.   1 root   root      7 Feb 11 13:47 lib -> usr/lib
...
```

Easy transition with **alias docker=podman**

- Some commands are docker-only (e.g., swarm, container-update)
- Some commands are podman-only
  - Podman supports health checks (running containers != healthy container)
  - Podman supports pods on the CLI (e.g., podman-pod-create)
  - Podman supports K8s yaml via podman-play-kube
    - Local K8s development without a cluster
    - Easy transition from and to K8s
  - Podman supports mounting the container rootfs via podman-mount
  - podman-image-tree for printing layer hierarchy, and more

Podman ABC

- Supports rootless containers since day 1
- It is not running as a daemon
  - Traditional fork-exec model
  - Improved security (reduced attack vector, adheres to security model, audit logging)
  - Covers additional use cases
- Remote client for Linux, Windows and Mac OS
  - Implemented via VARLINK.org
  - Varlink API can also be used for third-party applications (C, Go, Python, Java, Rust, bash)
  - Used in COCKPIT-PROJECT.org to manage containers in the browser
- Focus on OCI standards and open development
- Shares components with sibling projects (CRI-O, Buildah, Skopeo)

podman

Podman Resources

- Upstream development and community
  - github.com/containers/libpod
  - #podman of Freenode
  - podman@lists.podman.io
  - podman.io
- Demos
  - github.com/containers/demos
- Available on *most* Linux distributions
  - Red Hat Enterprise Linux, Fedora
  - openSUSE, Manjaro, Gentoo
  - Archlinux, Ubuntu, Debian (soon)

podman

Sylabs.io
Container Science

Singularity Runtime - 5 min

Michael Bauer - HPC Container Workshop ISC19

20 June 2019

CRITICAL MASS

Singularity is the open source container runtime of choice for

Artificial Intelligence, Compute Driven Analytics, Data Science…

- Millions of container runs per day
- With more than 40,000 users
- On millions of cores
- Across x86, ARM and POWER architectures

- Singularity voluntary registry, March 2019

HPC Wire Editors Choice Awards:

- 2016: Top products to watch
- 2017: Top products to watch
- 2017: Best HPC Programming Tool/Tech
- 2018: Best HPC Programming Tool/Tech
- 2018: Top Product to Watch

SINGULARITY USERS

VE RI TAS HARVARD UNIVERSITY

BERKELEY LAB

VETENSKAP OCH KONST KUNGL TEKNISKA HÖGSKOLAN

CINECA SCAI SuperComputing Applications and Innovation

PSC PITTSBURGH SUPERCOMPUTING CENTER

Research IT Advancing Research@Berkeley

DEPARTMENT OF DEFENSE UNITED STATES OF AMERICA

NYU

GSI

UNIVERSITAS STUDIORUM MEDIOLANENSIS UNIVERSITÀ DEGLI STUDI DI MILANO

FLORIDA STATE UNIVERSITY 1851 FLORIDA STATE UNIVERSITY

Open Science Grid

TACC TEXAS ADVANCED COMPUTING CENTER

Fermilab 50 Years of Discovery

Stanford University

RStor

Dartmouth

NIH National Institutes of Health

M UNIVERSITY OF MICHIGAN

DELL

Melbourne Bioinformatics BIOINFORMATICS • DATA SERVICES • INFRASTRUCTURE, FOR LIFE SCIENCES TODAY

USC University of Southern California Information Sciences Institute

Ohio Supercomputer Center An OH·TECH Consortium Member

IT4Innovations national01$#&0 supercomputing center@#01%101

Dartmouth

Hewlett Packard Enterprise

WESTGRID

THE UNIVERSITY OF IOWA

intel

NSC

1884 Memorial Sloan Kettering Cancer Center

ITWM Fraunhofer Institut Techno- und Wirtschaftsmathematik

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

VANDERBILT UNIVERSITY

UNIVERSITY TECHNOLOGY CASE WESTERN RESERVE UNIVERSITY

MARCC

American Museum of Natural History

INDIANA UNIVERSITY

Argonne NATIONAL LABORATORY

Queen Mary University of London

DTU Fysik Institut for Fysik

Lenovo

compute canada | calcul canada

UPPSALA UNIVERSITET

NVIDIA

UiO : University of Oslo

WASHINGTON STATE UNIVERSITY

amazon webservices

UTSouthwestern Medical Center Lyda Hill Department of Bioinformatics

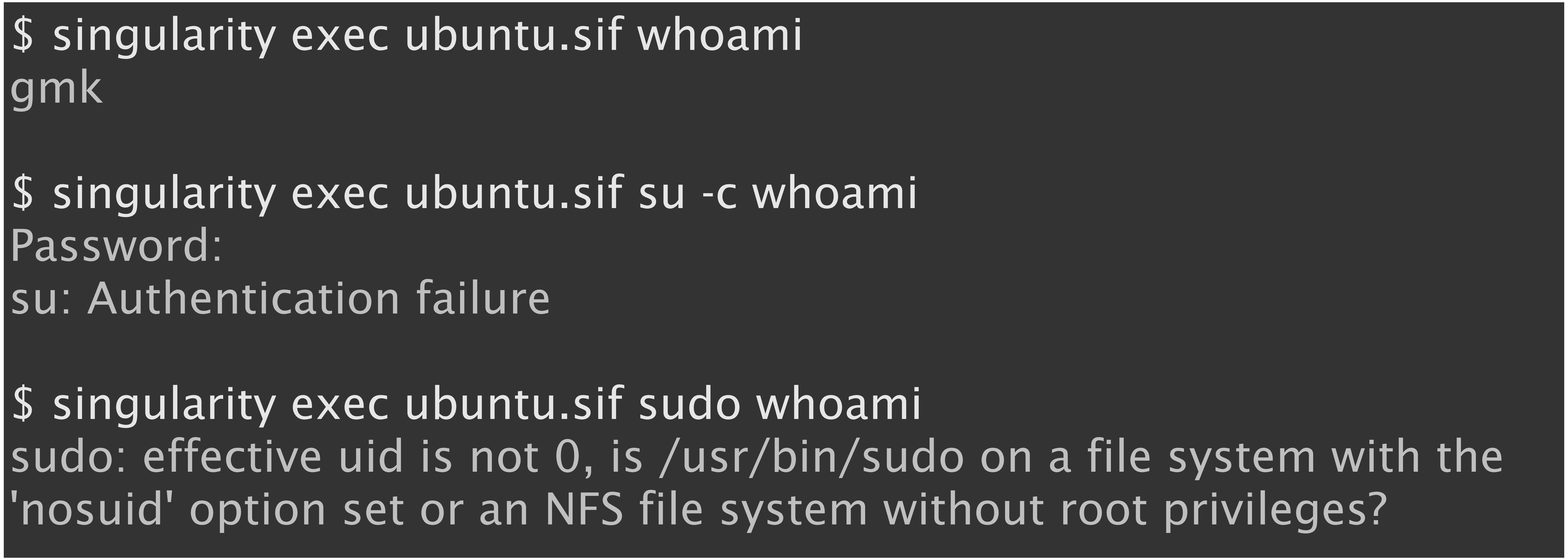
BioHPC

Center for High-Performance Computing @ the University of Utah

U.S. DEPARTMENT OF ENERGY Office of Science

Sylabs.io Container Science

BLOCKING ESCALATION

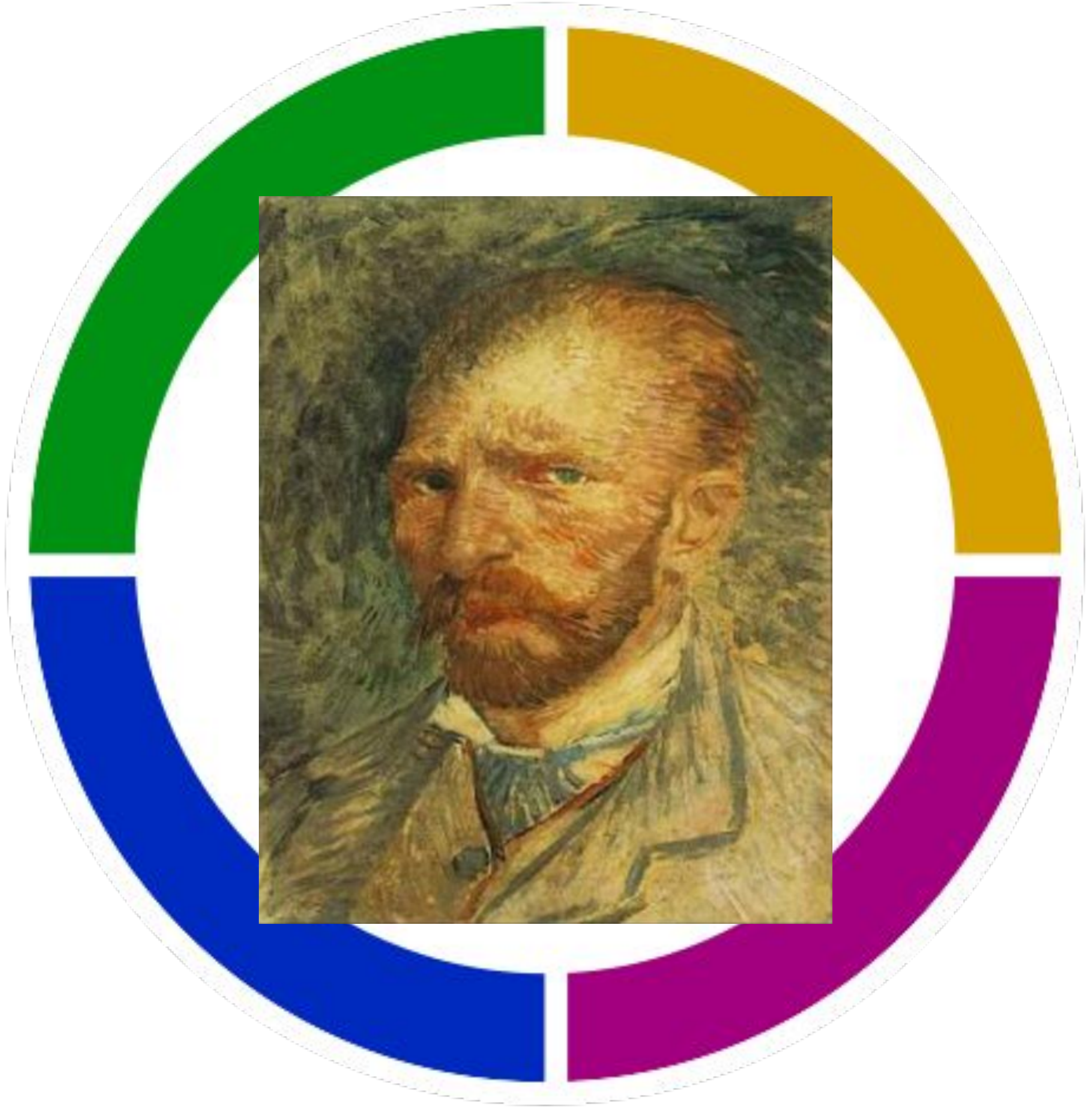


```
$ singularity exec ubuntu.sif whoami
gmk

$ singularity exec ubuntu.sif su -c whoami
Password:
su: Authentication failure

$ singularity exec ubuntu.sif sudo whoami
sudo: effective uid is not 0, is /usr/bin/sudo on a file system with the
'nosuid' option set or an NFS file system without root privileges?
```

Singularity blocks privilege escalation; once in side the container the user is always themselves

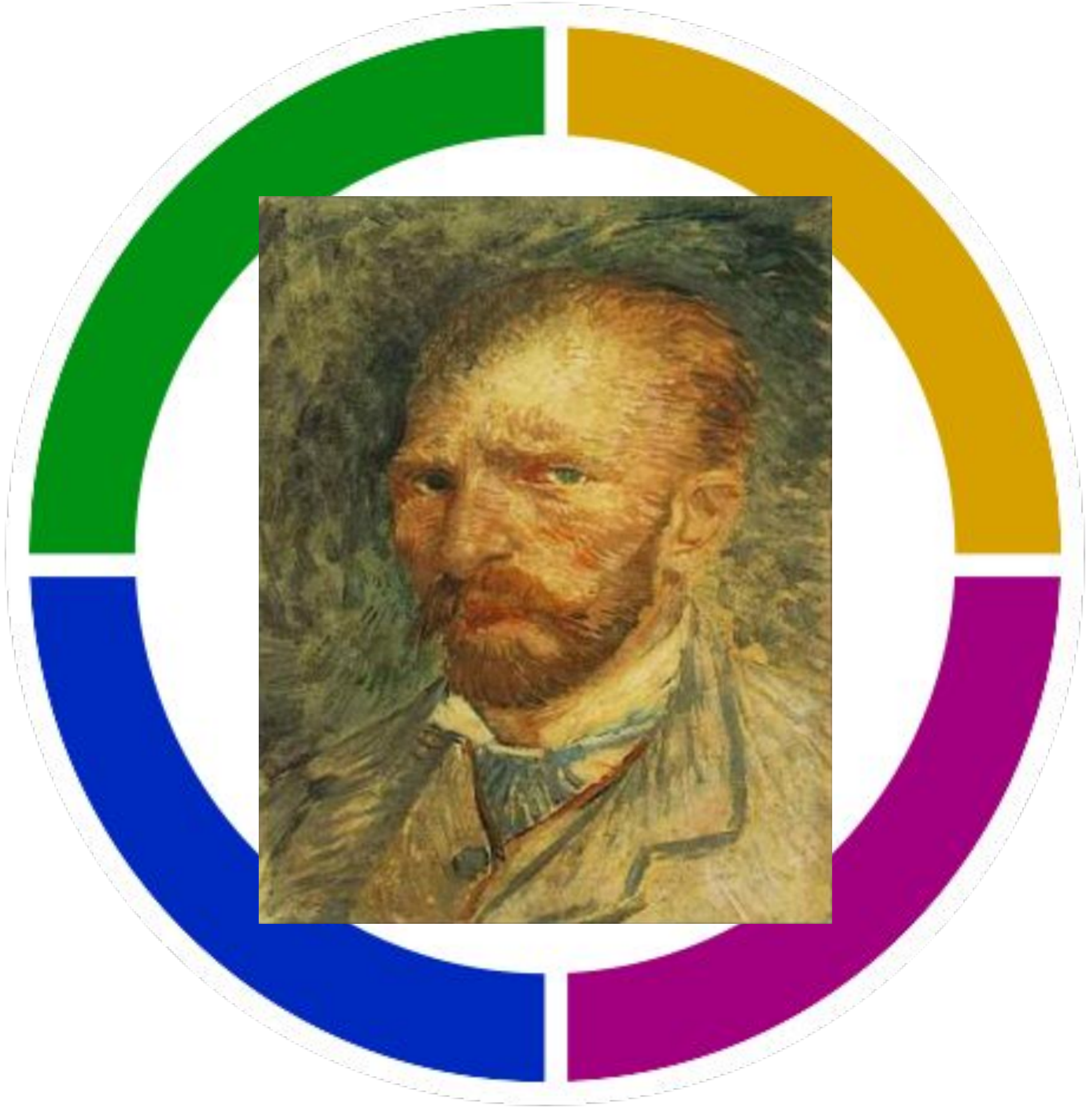
REPRODUCIBILITY

With Singularity, you get verifiable reproducibility



SHA:
5f09a35a642a68c467bf230f5e5ea3218e4177a0

=



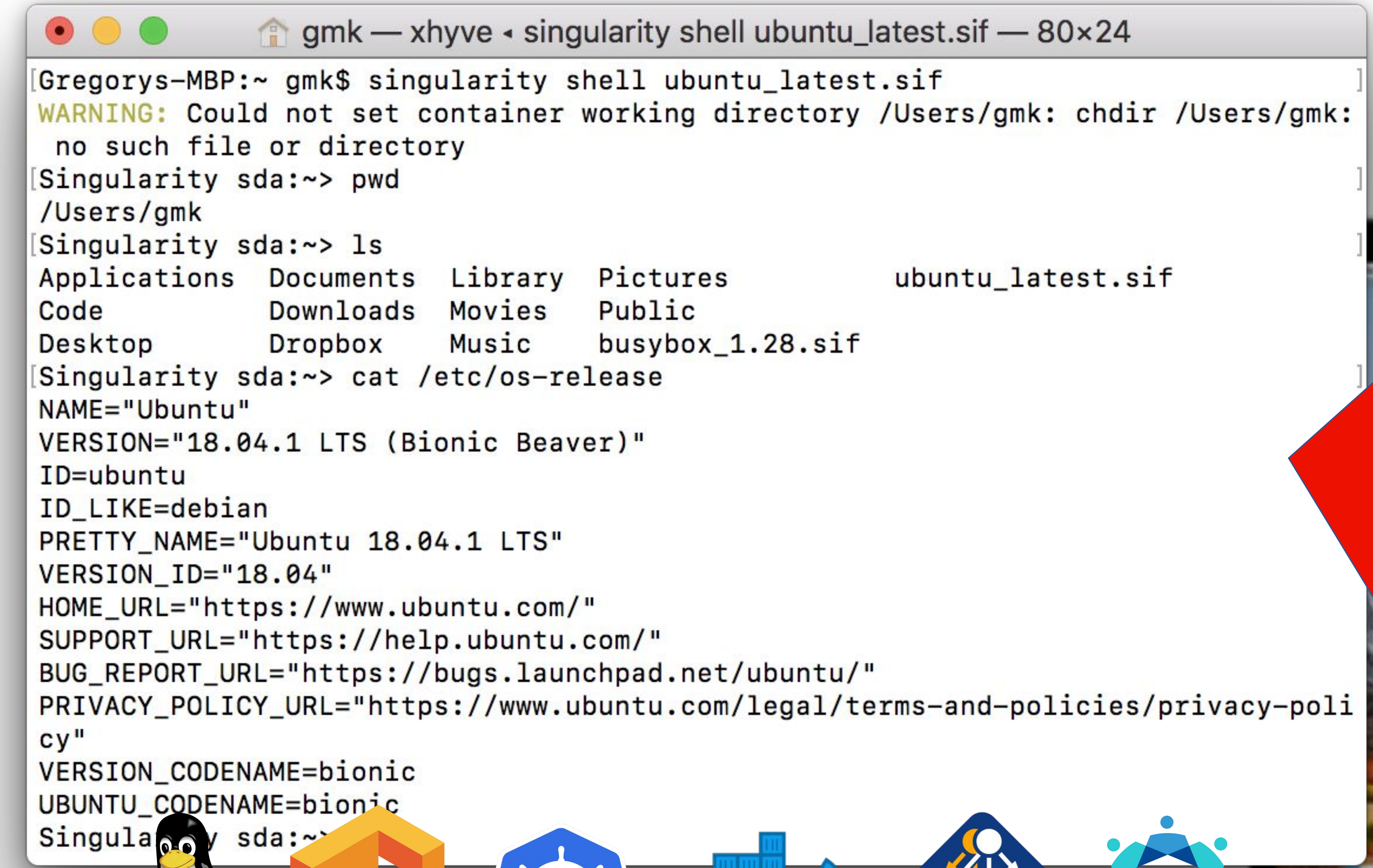
SHA:
5f09a35a642a68c467bf230f5e5ea3218e4177a0

Accessing Host GPU - TensorFlow

```
$ singularity exec --gpu=$(platform) docker://tensorflow/tensorflow 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()
$
```

Singularity Desktop

Singularity on MacOS



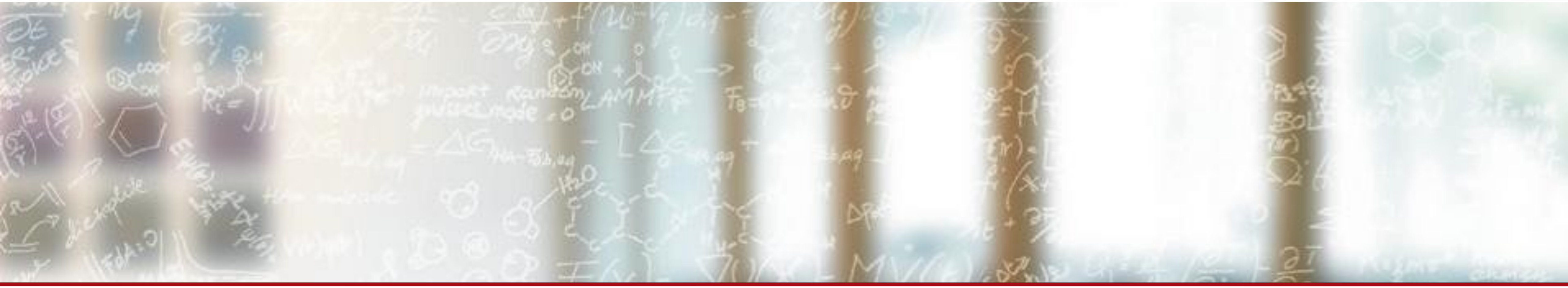
```
Gregorys-MBP:~ gmk$ singularity shell ubuntu_latest.sif
WARNING: Could not set container working directory /Users/gmk: chdir /Users/gmk:
 no such file or directory
Singularity sda:~> pwd
/Users/gmk
Singularity sda:~> ls
Applications  Documents  Library  Pictures         ubuntu_latest.sif
Code          Downloads  Movies   Public
Desktop       Dropbox    Music    busybox_1.28.sif
Singularity sda:~> cat /etc/os-release
NAME="Ubuntu"
VERSION="18.04.1 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.1 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-poli
cy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic
Singularity sda:~>
```

Singularity Desktop

Alpha released at #SUG19

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Sarus - An OCI-compliant container engine for HPC

HPCW 2019: 5th High Performance Containers Workshop
Lucas Benedicic, CSCS

June 20th, 2019

Comparison with existing solutions

	Suitable for HPC	**Pluggable vendor support** (standard OCI hooks)	**User experience**	**Admin experience**	**Maintenance effort**
Docker	🟥	🟥	🟩	🟨	🟩 +
Singula			🟧	🟩 +	🟩 +
Charlie			🟧	🟩 +	🟩 +
Shifter			🟨	🟧 -	🟧 -
LXC			🟧	🟧 -	🟩 +
runc	🟨	🟩 ++	🟥 --	🟩 ++	🟩 +
Sarus	🟩 +				

- **Suitable for HPC**
  - Single squashfs image (parallel filesystem friendly)
  - Image loop mount + RAM filesystem (fast image accesses)
  - WLM compatible
  - Native MPI support
  - Native GPU support

Comparison with existing solutions

	Suitable for HPC	**Pluggable vendor support** (standard OCI hooks)	**User experience**	**Admin experience**	**Maintenance effort**
Docker	--	🟥	🟩	🟨	+
Singularity	+			🟩	+
Charliecloud	+			🟩	+
Shifter	+			🟧	-
LXC	-			🟧	+
runc	🟨	🟩	--	++	+
Sarus	+	++			

- **Pluggable vendor support**
  - OCI hooks support (runc)
  - NVIDIA Container Runtime Hook

Comparison with existing solutions

	Suitable for HPC	Pluggable vendor support (standard OCI hooks)	User experience	Admin experience	Maintenance effort
Docker	--	--			
Singularity	+	-			
Charliecloud	+	--			
Shifter	+	-			
LXC	-	-			
runc	🟨	++		++	+
Sarus	+	++	++		

- **User Experience**
  - Docker-like CLI
  - Docker Hub integration
  - OverlayFS (writable container filesystem)
  - Preserve identity and file permissions

Comparison with existing solutions

	Suitable for HPC	Pluggable vendor support (standard OCI hooks)	User experience	Admin experience	Maintenance effort
Docker	--	-			+
Singularity	+				+
Charliecloud	+	-			+
Shifter	+				-
LXC	-				+
runc		++	--		+
Sarus	+	++	++	+	

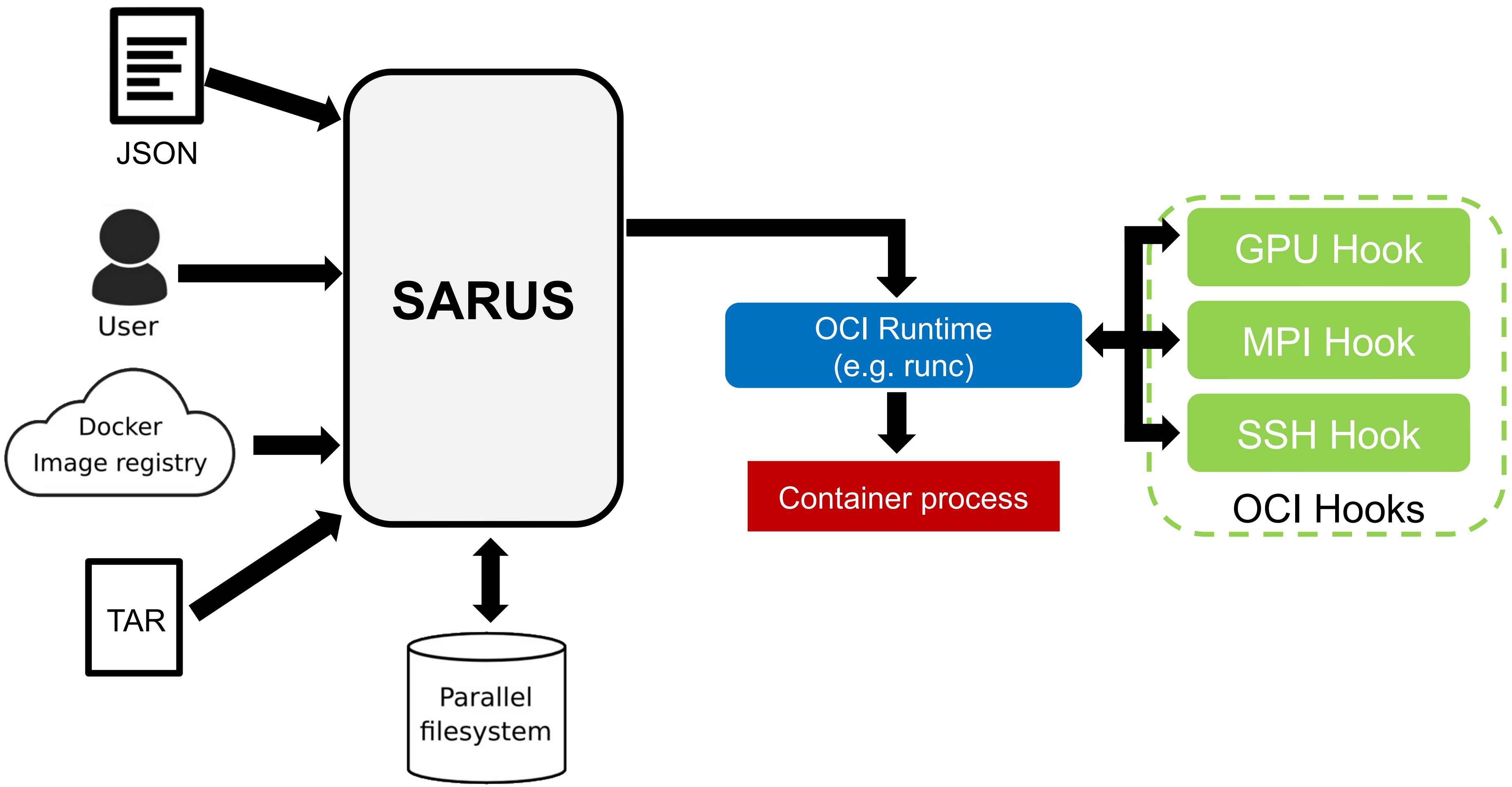
- **Admin experience**
  - Single executable binary (easy deployability)
  - Customize OCI hooks per system
  - Container isolation through **PID** and **runc**

Comparison with existing solutions

	Suitable for HPC	**Pluggable vendor support** (standard OCI hooks)	**User experience**	**Admin experience**	**Maintenance effort**
Docker	--	--	+		
Singularity	+	-			
Charliecloud	+	--	-		
Shifter	+	-			
LXC	-	-			
runc		++	--	++	
Sarus	+	++	++	+	+

- **Maintenance effort**
  - Reuse **runc** as the core runtime
  - Reuse other OCI-compliant software
  - Well tested (unit test coverage 84%)

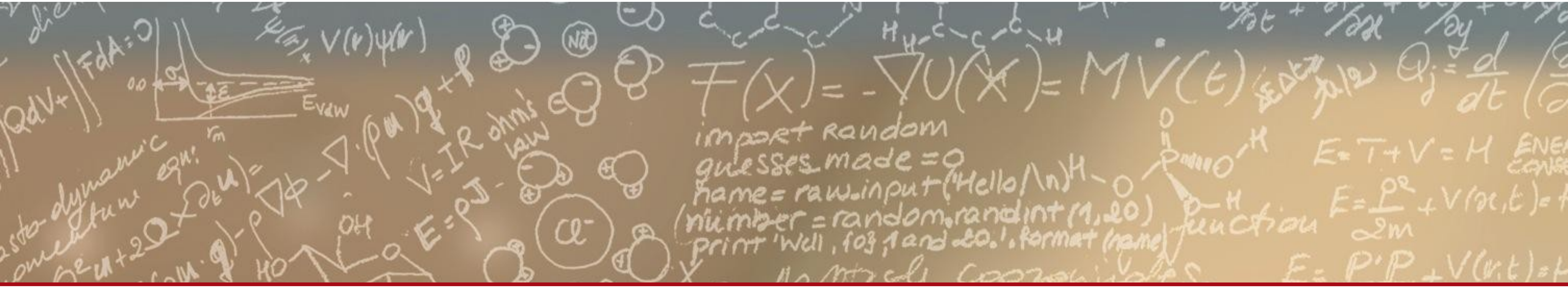
Architecture overview



Conclusion

Sarus is a container engine for HPC, compliant with open standards, featuring:

- Transparent native performance through OCI hooks
- Consistent UX with Docker: small learning curve
- Enables use of standard, open, upstream components on HPC systems
- Extensible architecture encourages vendor engagement and improves maintainability



Thank you for your attention.