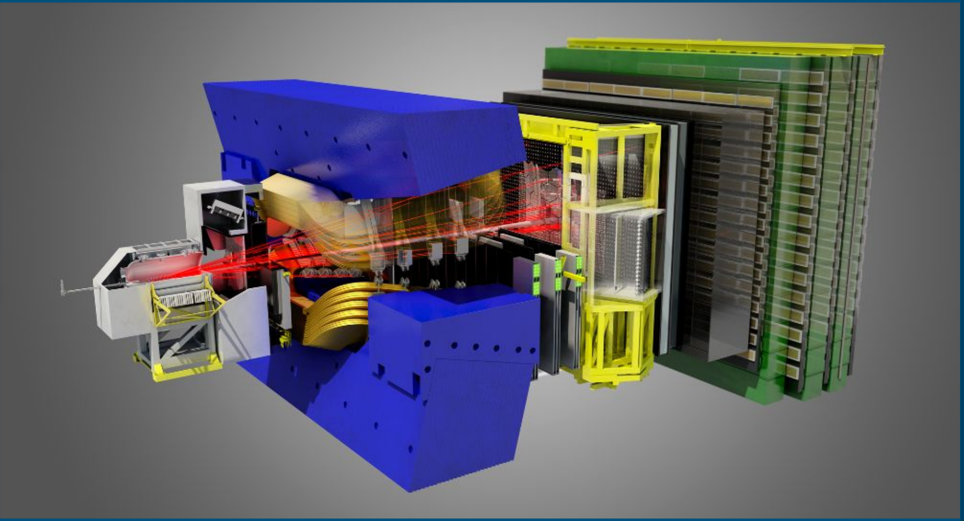
Containers@LHCb

ISGC Containers Workshop
Ben Couturier (for the LHCb Computing team)

LHCb Collaboration

~1200 members,
69 Institutes,
16 Countries



Containers in LHCb

Appealing way to package/run experimental software

- Need to build/run on SLC5/SLC6/Centos7
- S.Binet prepared images in since 2014

Boundary conditions:

LHCb application stack is rather large

- 12GB for the application software
- slc6-build image: ~1.2GB

Integration with CVMFS is therefore crucial

- Either inside the docker image (not practical)
- Or on the system, shared between all images

Multiple teams interested

The computing team saw potential early on

- in line with microservices and builds
- Using production software

Analysts are also very keen, for different reasons

- Easy way to gather/configure all the tools needed for analysis
- Really helps reproducibility, in conjunction with gitlab/gitlabCI

Containers for developers

Containers for continuous build environment

Container with correct OS started on demand by the Jenkins jobs

- Great way to decouple build VMs from the builds
- Extremely useful for old stacks…

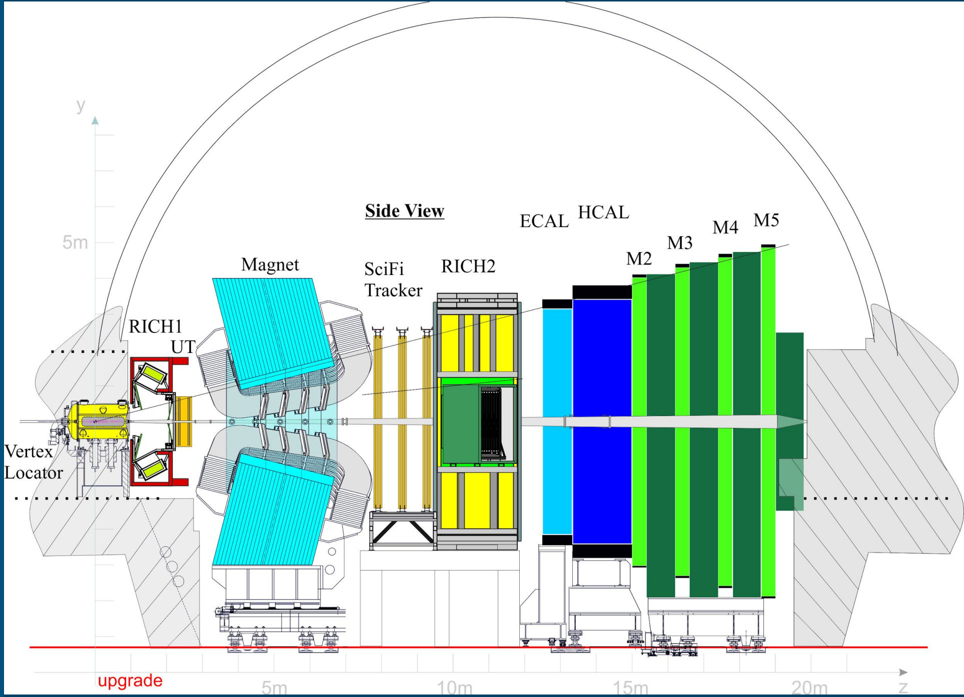
But not without consequences

- Restarting docker daemon can have disastrous effects
- Ditto for Jenkins agent

Containers as development environment

Practical way to allow development on developers' laptops

- Used in the LHCb upgrade hackathons
- Provided CVMFS is already mounted
- Tricks needed for graphical applications…



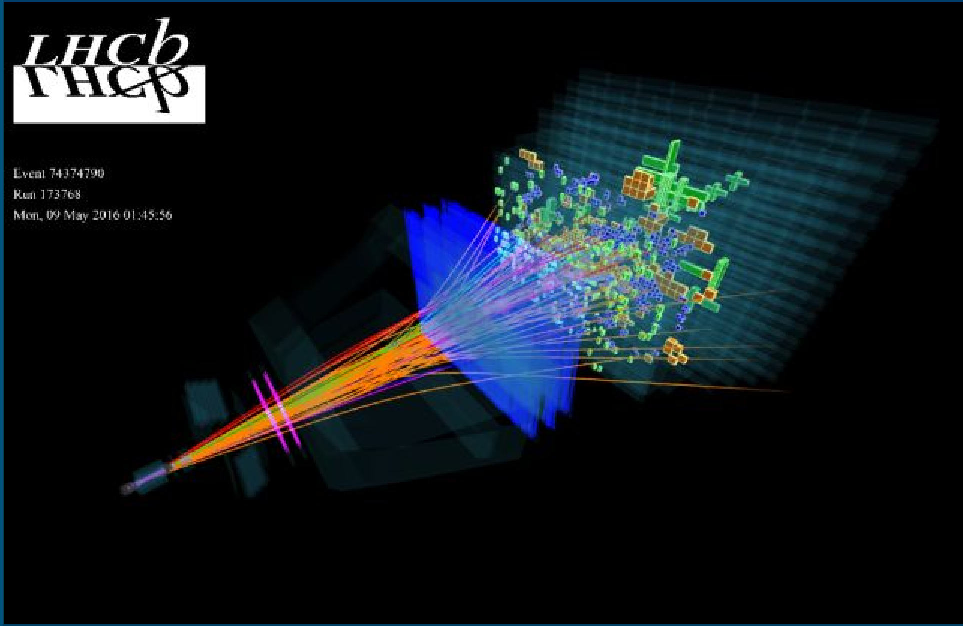
Containers as development environment

Unfortunately containers cannot be used on shared clusters

- Is Openshift a solution ?

Not convincing on MacOS yet

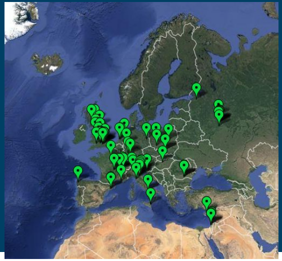
- I/O performance issues
- CernVM probably more convincing still



Rerunning productions

Using the LHCb Software Preservation DB

- Can rerun any LHCb data processing
- Uses the Preservation DB to choose software version
- Runs on the appropriate image, loading applications from CVMFS



Reproducible Analysis

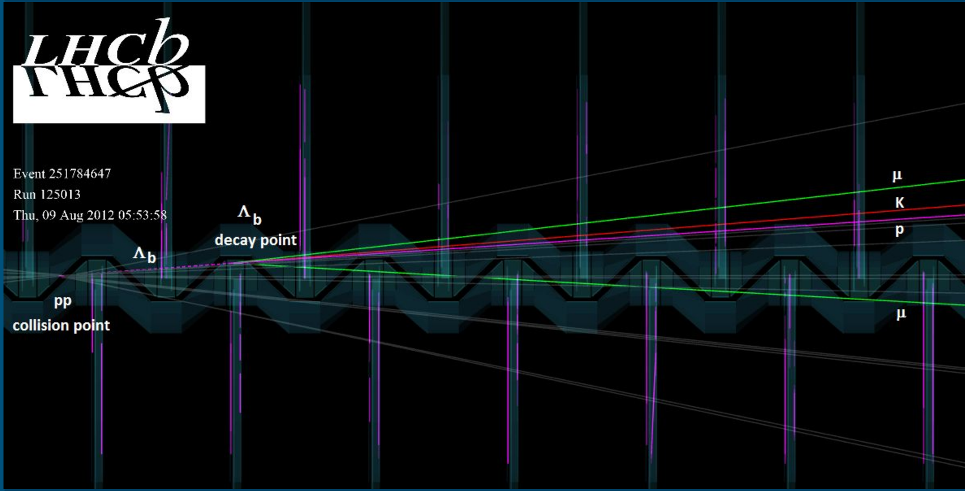
Reproducible Research@LHCb

Different issue than production

- Analysis software not necessarily on CVMFS
- Wider range of tools and methods
- Does not run on same data volumes

Containers are a key enabler of "Continuous Analysis"

- Used in coordination with version control systems (git/gitlab in our case) and continuous integration

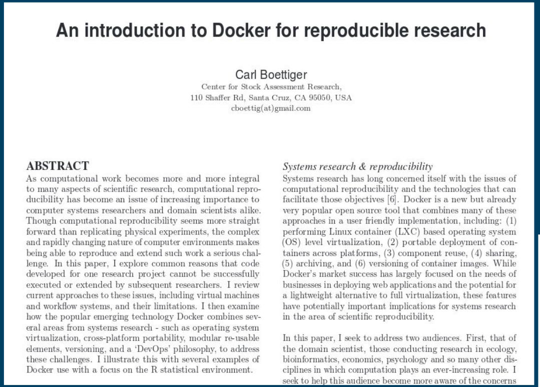


Huge interest in the research community

Have tried to follow various initiatives...

Obviously NOT an exhaustive list!

- Biomed researchers are very interested
- Way to run custom code on HPC clusters



An introduction to Docker for reproducible research

Carl Boettiger
Center for Stock Assessment Research,
110 Shaffer Rd, Santa Cruz, CA 95050, USA
cboettig(at)gmail.com

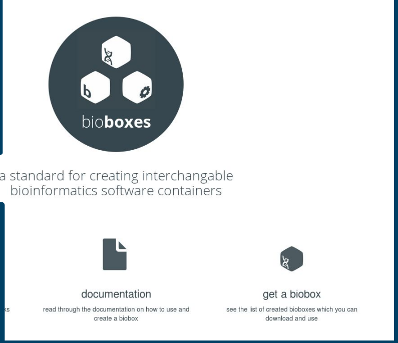
ABSTRACT

As computational work becomes more and more integral to many aspects of scientific research, computational reproducibility has become an issue of increasing importance to computer systems researchers and domain scientists alike. Though computational reproducibility seems more straight forward than replicating physical experiments, the complex and rapidly changing nature of computer environments makes being able to reproduce and extend such work a serious challenge. In this paper, I explore common reasons that code developed for one research project cannot be successfully executed or extended by subsequent researchers. I review current approaches to these issues, including virtual machines and workflow systems, and their limitations. I then examine how the popular emerging technology Docker combines several areas from systems research - such as operating system virtualization, cross-platform portability, modular re-usable elements, versioning, and a 'DevOps' philosophy, to address these challenges. I illustrate this with several examples of Docker use with a focus on the R statistical environment.

Systems research & reproducibility

Systems research has long concerned itself with the issues of computational reproducibility and the technologies that can facilitate those objectives [6]. Docker is a new but already very popular open source tool that combines many of these approaches in a user friendly implementation, including: (1) performing Linux container (LXC) based operating system (OS) level virtualization, (2) portable deployment of containers across platforms, (3) component reuse, (4) sharing, (5) archiving, and (6) versioning of container images. While Docker's market success has largely focused on the needs of businesses in deploying web applications and the potential for a lightweight alternative to full virtualization, these features have potentially important implications for systems research in the area of scientific reproducibility.

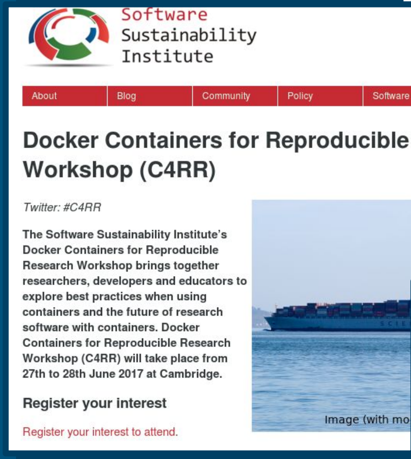
In this paper, I seek to address two audiences. First, that of the domain scientist, those conducting research in ecology, bioinformatics, economics, psychology and so many other disciplines in which computation plays an ever-increasing role. I seek to help this audience become more aware of the concerns



bioboxes

a standard for creating interchangable bioinformatics software containers

documentation
read through the documentation on how to use and create a biobox
get a biobox
see the list of created bioboxes which you can download and use



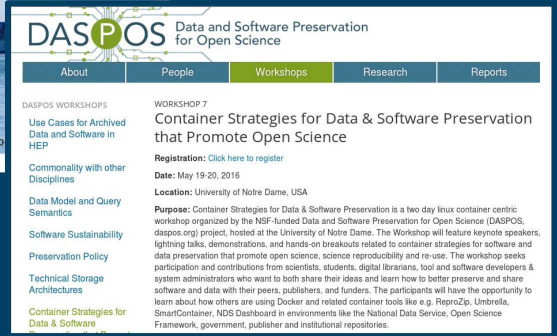
Docker Containers for Reproducible Research Workshop (C4RR)

Twitter: #C4RR

The Software Sustainability Institute's Docker Containers for Reproducible Research Workshop brings together researchers, developers and educators to explore best practices when using containers and the future of research software with containers. Docker Containers for Reproducible Research Workshop (C4RR) will take place from 27th to 28th June 2017 at Cambridge.

Register your interest

Register your interest to attend.



DASPOS Data and Software Preservation for Open Science
About
People
Workshops
Research
Reports
DASPOS WORKSHOPS
Use Cases for Archived Data and Software in HEP
Commonality with other Disciplines
Data Model and Query Semantics
Software Sustainability
Preservation Policy
Technical Storage Architectures
Container Strategies for Data & Software

WORKSHOP 7

Container Strategies for Data & Software Preservation that Promote Open Science

Registration: Click here to register

Date: May 19-20, 2016

Location: University of Notre Dame, USA

Purpose: Container Strategies for Data & Software Preservation is a two day linux container centric workshop organized by the NSF-funded Data and Software Preservation for Open Science (DASPOS, daspos.org) project, hosted at the University of Notre Dame. The Workshop will feature keynote speakers, lightning talks, demonstrations, and hands-on breakouts related to container strategies for software and data preservation that promote open science, science reproducibility and re-use. The workshop seeks participation and contributions from scientists, students, digital librarians, tool and software developers & system administrators who want to both share their ideas and learn how to better preserve and share software and data with their peers, publishers, and funders. The participants will have the opportunity to learn about how others are using Docker and related container tools like e.g. ReproZip, Umbrella, SmartContainer, NDS Dashboard in environments like the National Data Service, Open Science Framework, government, publisher and institutional repositories.

Data Analysis tools following suit

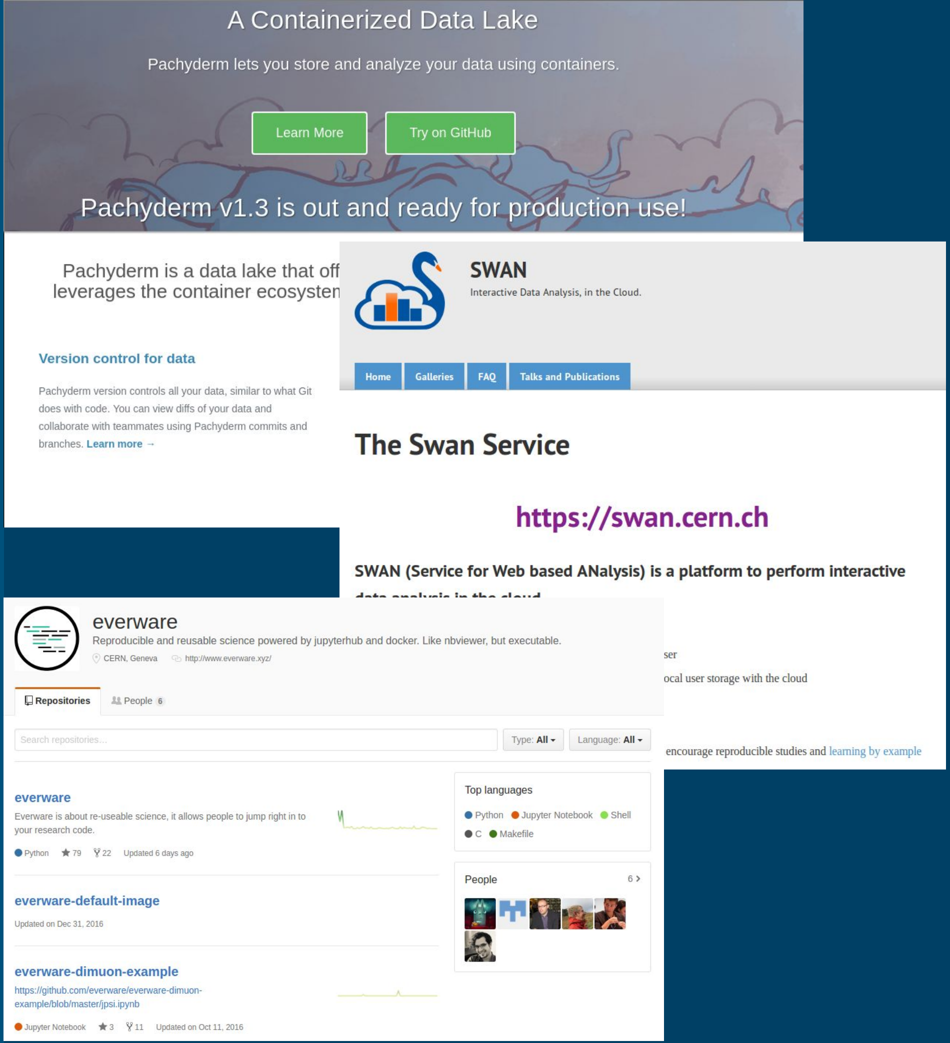
Containers are the enabling technology for complex environments in pipelines:

- e.g. www.pachyderm.io

But also for interactive analysis:

- https://github.com/everware
- https://cern.ch/swan
- [...]

How do we run containerized analysis on our compute resources ?



A Containerized Data Lake

Pachyderm lets you store and analyze your data using containers.

Pachyderm v1.3 is out and ready for production use!

Pachyderm is a data lake that off
leverages the container ecosysten

Version control for data

Pachyderm version controls all your data, similar to what Git does with code. You can view diffs of your data and collaborate with teammates using Pachyderm commits and branches. Learn more →

SWAN

Interactive Data Analysis, in the Cloud.

The Swan Service

https://swan.cern.ch

SWAN (Service for Web based ANalysis) is a platform to perform interactive data analysis in the cloud

ser

ocal user storage with the cloud

encourage reproducible studies and learning by example

everware

Reproducible and reusable science powered by jupyterhub and docker. Like nbviewer, but executable.

CERN, Geneva http://www.everware.xyz/

everware

Everware is about re-useable science, it allows people to jump right in to your research code.

Python ★ 79 22 Updated 6 days ago

everware-default-image

Updated on Dec 31, 2016

everware-dimuon-example

https://github.com/everware/everware-dimuon-example/blob/master/jpsi.ipynb

Jupyter Notebook ★ 3 11 Updated on Oct 11, 2016

Top languages

● Python ● Jupyter Notebook ● Shell ● C ● Makefile

People 6

Containers
v.s.
VMs

Maybe we don’t have to choose

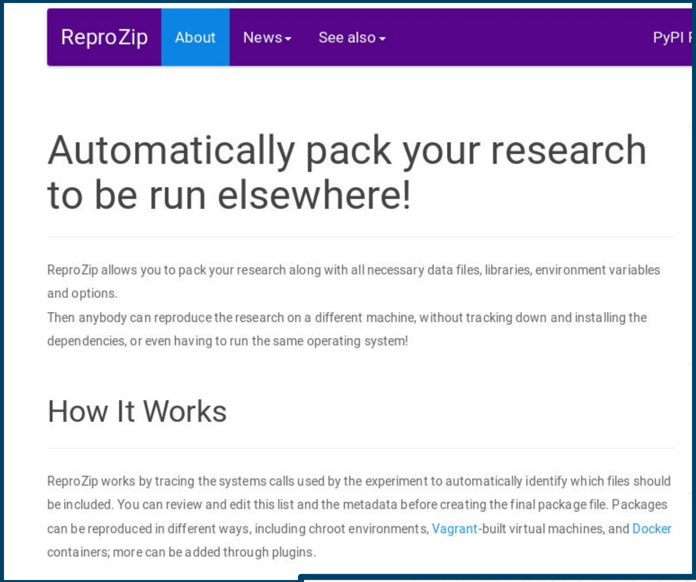
- Some tools can use both e.g. ReproZip

They can be used in conjonction anyway

- Openstack to start VMs running dockers images...

Customizing VM images is not necessarily so complicated

- Vagrant can be used with VMs

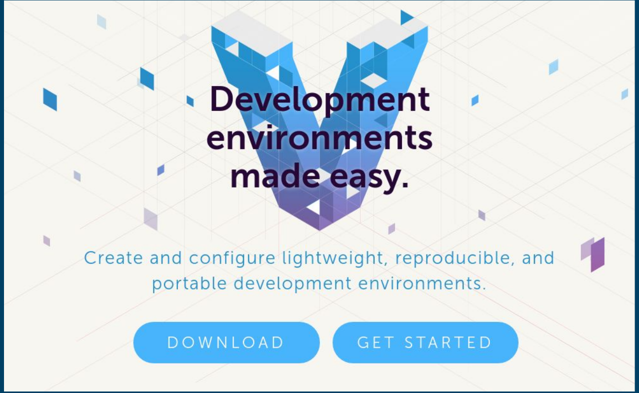


Automatically pack your research to be run elsewhere!

ReproZip allows you to pack your research along with all necessary data files, libraries, environment variables and options.
Then anybody can reproduce the research on a different machine, without tracking down and installing the dependencies, or even having to run the same operating system!

How It Works

ReproZip works by tracing the systems calls used by the experiment to automatically identify which files should be included. You can review and edit this list and the metadata before creating the final package file. Packages can be reproduced in different ways, including chroot environments, Vagrant-built virtual machines, and Docker containers; more can be added through plugins.



Create and configure lightweight, reproducible, and portable development environments.

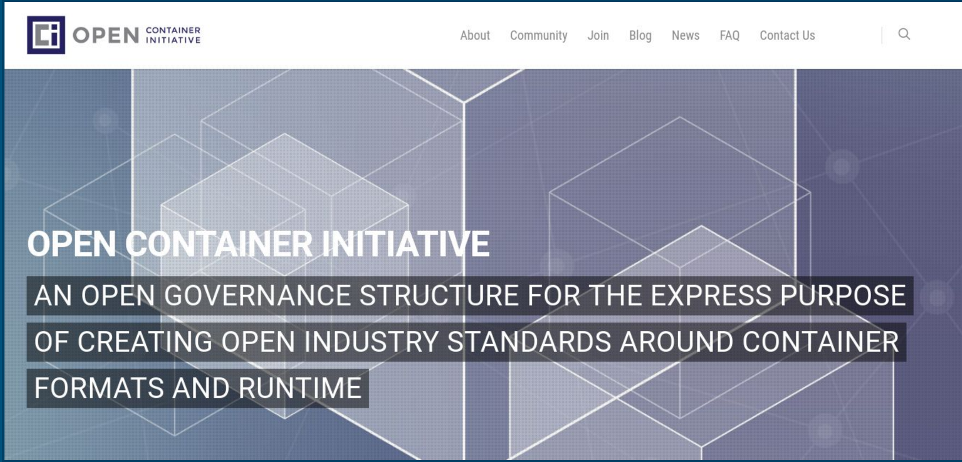
Long term preservation of containers ?

How long do we need to run them for ?

- In the case of LHCb, we have to re-run old trigger versions for the duration of the experiment

Are containers more/less preservable than VMs?

- **Would standards help ?** https://www.opencontainers.org/



OPEN CONTAINER INITIATIVE

AN OPEN GOVERNANCE STRUCTURE FOR THE EXPRESS PURPOSE OF CREATING OPEN INDUSTRY STANDARDS AROUND CONTAINER FORMATS AND RUNTIME

Exciting times!

- Very active field, loads of exciting developments, many questions open
- How to make sure our users can make full use of containers ?
- How to articulate VMs and containers ?
- How to organize all the images used ?
- What about security aspects ?

We are not alone, need for collaboration with other fields on the topic