

A decorative graphic in the top-left corner consisting of several overlapping, semi-transparent squares of varying shades of blue, arranged in a roughly triangular pattern pointing towards the top-left.

LHCb perspective on containers

Andrew McNab
University of Manchester,
GridPP, and LHCb

Containers as logical machines

- LHCb is running at two sites which use LHCb containers
 - Andrew Lahiff's system at RAL, and Skygrid at Yandex
 - Both use (different) containers derived from LHCb DIRAC VMs
- We've developed a generic LHCb container definition based on this experience
 - Uses Docker
 - Uses cernvm root image (ie via cvmfs)
 - LHCb cvmfs and /init script to run inside the container also provided via volumes
 - Uses a format which will be supported by Vac and (inside a generic Docker VM) by Vcycle

Containers as glxec replacements

- LHCb are following Singularity discussions in WLCG Traceability and Isolation TF
- We believe it will be straightforward to add Singularity functionality to DIRAC
 - Already have support for glxec and sudo wrappers around payload jobs in the DIRAC pilot
 - Singularity-based wrapper can be added there
- We intend to test this in the LHCb DIRAC VMs
 - We currently use the sudo wrapper and run each payload in a dedicated unix user account in the VM
- We don't have a big requirement for Singularity to provide SL6 on CentOS 7 worker nodes
 - Uniformity is always easier though

Containers as a user job format

- There is a lot of interest from LHCb users in packaging their jobs in, say, Docker images
- Allows reuse of other people's code and management of what the user has changed
- Makes analyses more reproducible and easier to recreate in the future
- We could try to support this in LHCb DIRAC VMs where we have root and could run Docker
 - But it's a lot of work for something that can't be applied to grid/batch
 - Getting users to target Singularity might be a viable route though (still as Docker images?)