

LHCb Dockerized Build Environment

M. Belin, M. Clemencic, J. Closier and B. Couturier on behalf of the LHCb Core Software Team

Abstract

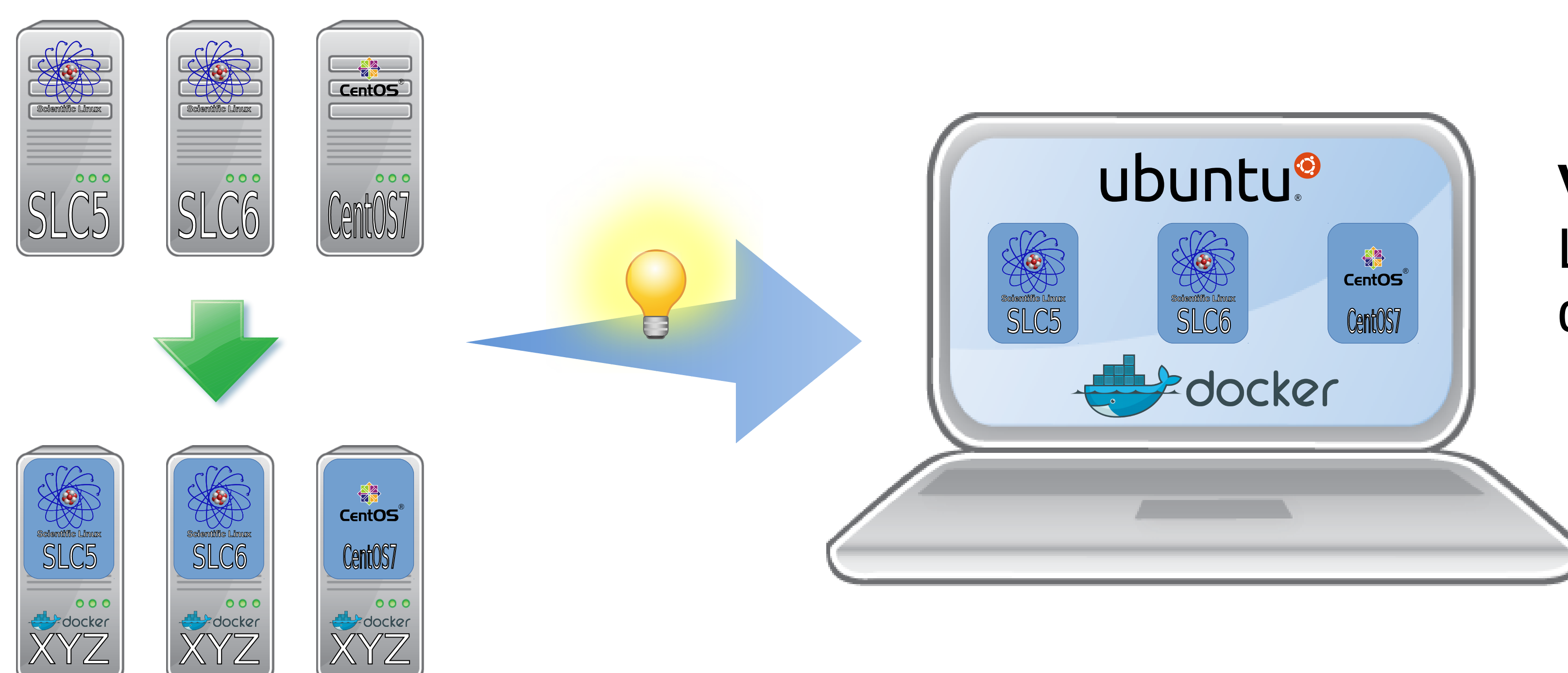
Used as lightweight virtual machines or as enhanced chroot environments, Linux containers, and in particular the Docker abstraction over them, are more and more popular in the virtualization communities.

The LHCb Core Software team decided to investigate how to **use Docker containers to provide stable and reliable build environments** for the different supported platforms, including the obsolete ones which cannot be installed on modern hardware, to be used in **integration builds, releases and by any developer**.

We present here the techniques and procedures set up to define and maintain the Docker images and how these images can be used to develop on modern Linux distributions for platforms otherwise not accessible.

Plan:
replace heterogeneous build cluster with homogeneous pool of machines.

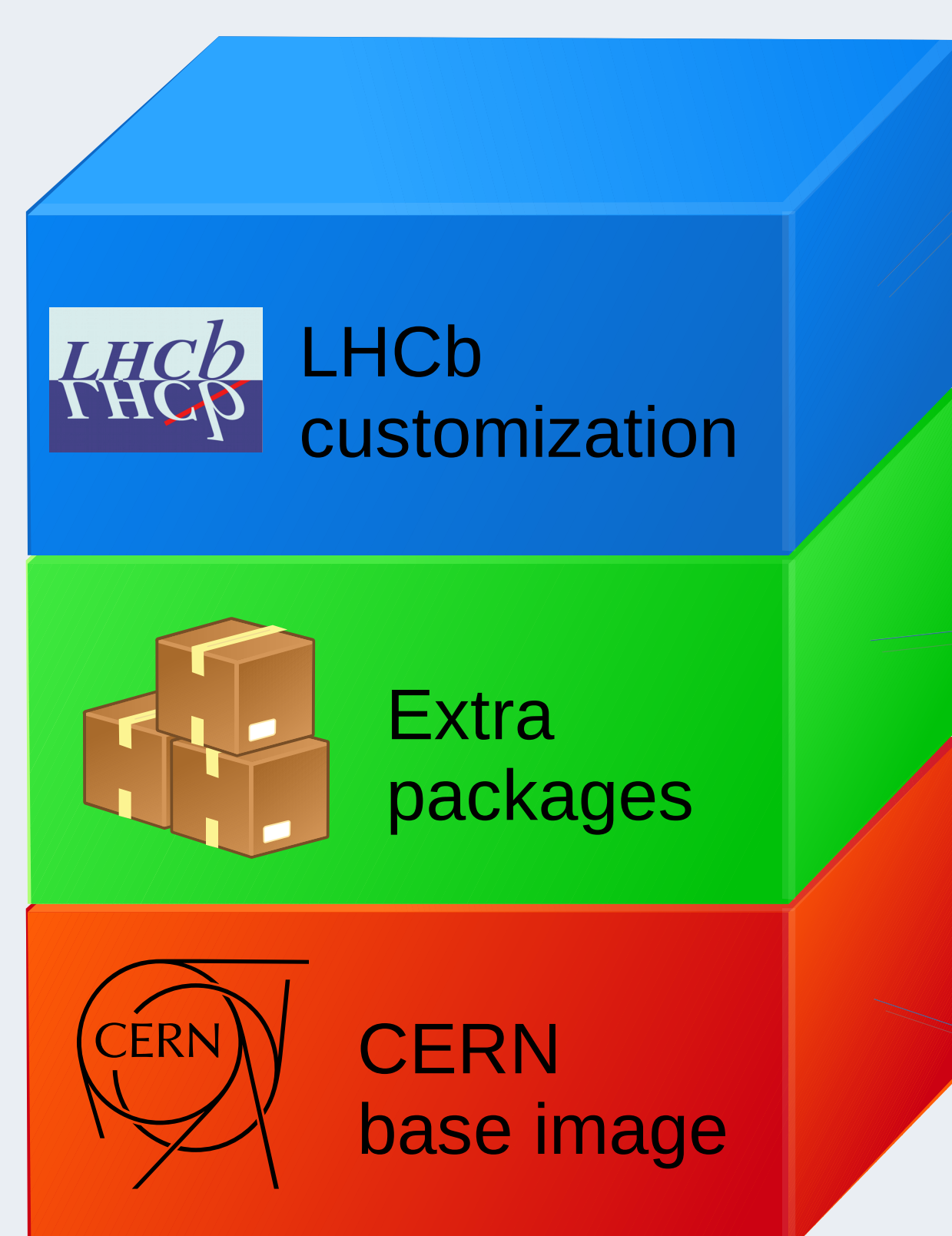
Target:
better distribute the build load.



Vision:
Lightweight portable development environment.

LHCb Docker images

- One per supported platform (SLC5, SLC6, CentOS7)
- Share workspace with host
- Unprivileged user
- Standard LHCb environment
- CVMFS from host or mounted internally
- Custom wrapper around "docker run"



Custom settings and scripts for improved user experience (LHCb environment, unprivileged user, workspace shared with host, ...)

Standard software packages required to build and run LHCb Software (Hep_OSlibs, g++, ...)

Minimal preconfigured images (system settings, software repositories, ...)

lb-docker-run

- Wraps "docker run"
- Simplified command line interface
 - Shared volumes
 - Workspace
 - Persistent user home
 - Share CVMFS mounts
 - Communicate with lb-su

LHCb customization

lb-su

- Inside the container, tunes environment as requested via lb-docker-run
- Switch to host user
 - Call LHCb login script
 - Mount CVMFS if needed

LHCb Nightly Builds

Being their primary goal, LHCb Docker images are currently **used in production** by the **LHCb Nightly Build System** to build and test LHCb Software on **all supported platforms**, using a **uniform cluster** of CentOS7 build machines.

LHCb Upgrade Hackathon

LHCb Docker images and the lb-docker-run tool have been used to provide a **well defined development environment with prebuilt software** projects that developers could modify. The prebuilt images and the uniformity of the environment made it possible to leverage on **incremental builds** to speed up the development during the hackathons organized to work on the software for the LHCb Upgrade.

Analysis Preservation

LHCb Docker images provide an ideal ground for Analysis Preservation, because they allow **execution of old versions of LHCb analysis code on modern systems** without the burden of hardware virtualization. Moreover, being developed for the continuous integration system of LHCb, they are validated using the existing software unit and integration tests.