

# Pre-GDB on Software Deployment

## Welcome and Introduction

---

Jakob Blomer and Alessandra Forti

# Welcome!

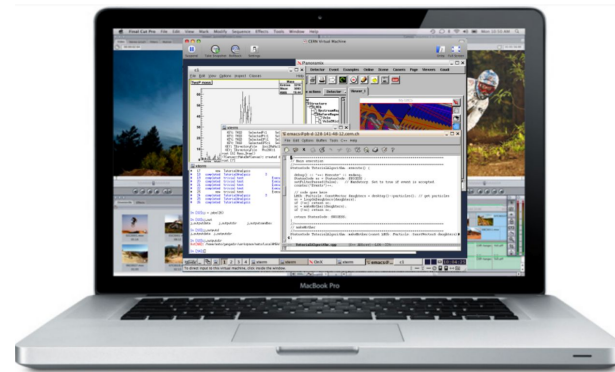
- We hope you're well and able to enjoy a cup of coffee or tea
- First remote only pre-GDB (April GDB was remote-only, too)
- Replacement for the [BoF](#) at the WLCG/HSF/OSG workshop
  
- Agenda for today
  - Morning session: librarians perspective from experiments and platform providers (HEP\_OSlibs and LCG stack)
  - Afternoon: infrastructure perspective: cvmfs, container technologies, k8s
  - 1pm to 3pm: lunch break
    - Time to attend the CERN COVID-19 update  
<https://indico.cern.ch/event/912581/>
- [Live notebook](#)
- Summary of this meeting at tomorrow's GDB  
<https://indico.cern.ch/event/813747/>



illustration is not binding

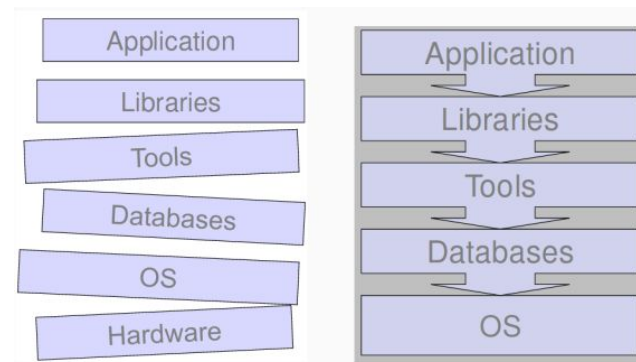
# Software Deployment: CernVM Experience

- The CernVM R&D project started in a world of grid installation jobs and when Amazon had just launched EC2
- The project was fundamentally about software deployment: *provide a portable platform for developing and running LHC applications*
- We still do this until today



Some useful principles emerged from the project

1. Horizontal integration: the application should define which bits and pieces are needed from underlying layers (OS, libraries, ...: *The Platform*)
2. Creation of the runtime environment from thin air, i.e. fine-granular on-demand distribution and caching
3. The environment for *developing applications* should be identical to the environment for *running applications*
4. Long term software preservation needs to be built-in, i.e. preservation effort needs to be part of the regular deployment activities



# Technology Toolbox

- CernVM-FS for the distribution of software binaries: production software, nightly builds, containers
- Container technology
  - Multiple popular container runtime engines: singularity, containerd, docker (developers)
  - Kubernetes predominant container orchestration tool, gains traction as a base layer for batch resources (<https://indico.cern.ch/event/739899>)
- Tools for building software stacks
  - Base layers from LCG stack and HEP\_OSlibs
  - Spack package manager, addresses many of the common build issues

At the same time: more complex software stacks (multiple architectures, external ML tools, ...) and more heterogeneous resources (grid and more)

**What would we ideally use for run 3 and 4?**

**Where do we need to invest?**

