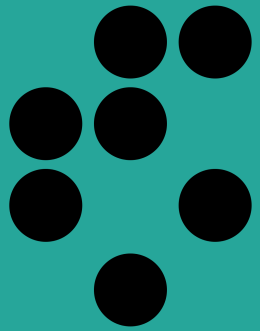


Singularity containers
A mini tutorial

—

Blaž Škrlj with the help of
Barbara Krašovec

The four levels of repositories one finds these days

1. Messy, undocumented code with *no dependency specifications*
2. Code with dependency specifications, *not versioned*
3. Versioned code, dependencies
4. **Code with a replicable environment**

(1) What is Singularity?

1. Assuming you wish to share some code with a colleague
2. What's the minimal effort she/he needs to undergo to run the code?

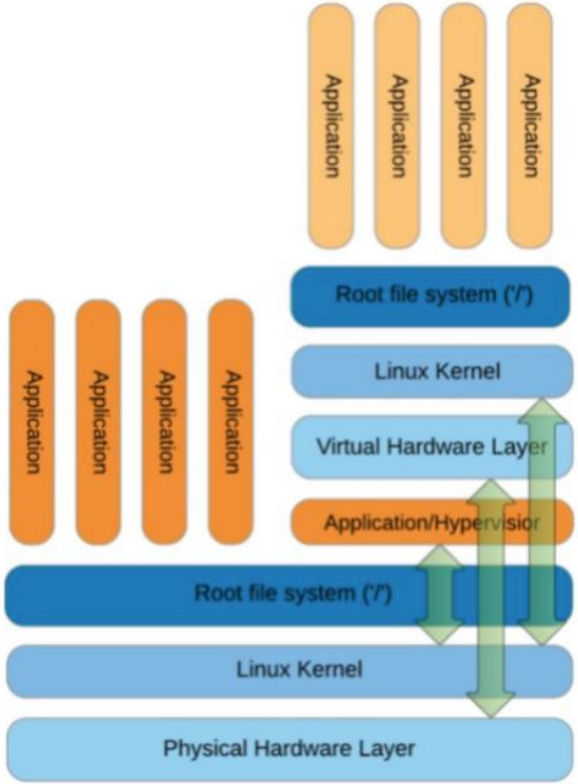
(2) What is Singularity?

1. It's an environment for **dockerization**. Specifically focused on **scientific** and **HPC** environments.

In other words:

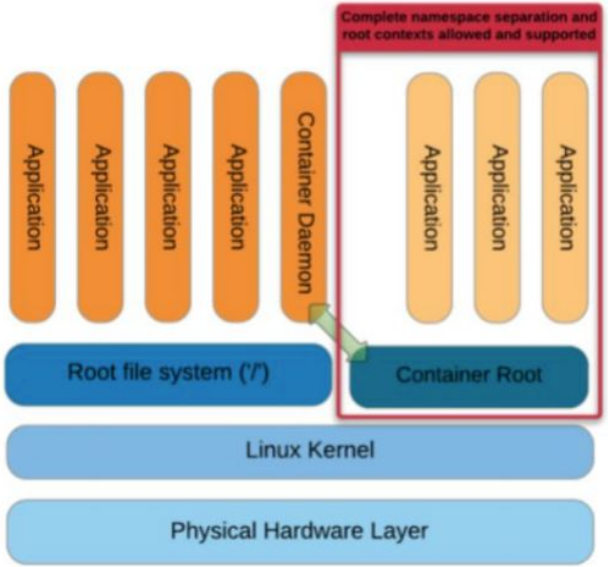
It's the engineering swiss knife suitable for the **fast-paced** (replicable) science**.**

Virtual Machine



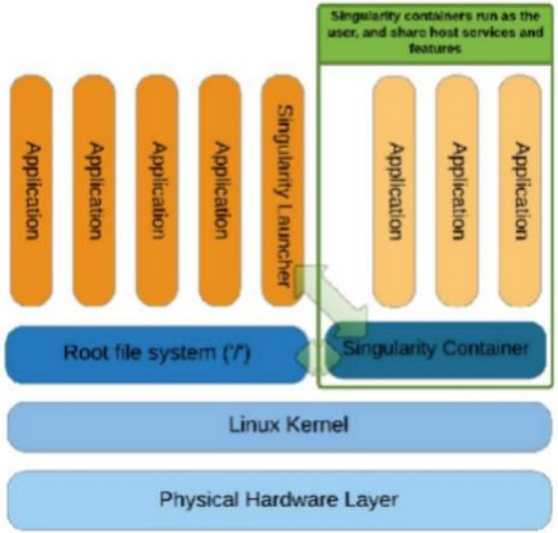
Docker

- Independent namespaces
- Root supported contexts



Singularity

- Host services and devices shared
- User access



https://www.bsc.es/support/PATC/2ndDAY/11:00-12:00_Containers-HPC.pdf

Summary

When to use it

Testing

- Unstable releases
- Proof of concept executions
- Punctual data preproces

Reproducibility

- Complex software stacks

When NOT to use it

Production

- Architecture aware software

  GROMACS, NAMD, CPMD...
- Frequent HPC applications

Not available soft?

- Request it to the support team

Detailed info at: https://sylabs.io/docs/

Extremely useful for

Deployment on multiple machines (HPC).

Quick replicable demos and handling dependency problems.

“Oh, this machine is not doing anything, let’s just run this experiment” - type of adventures

Tutorial time

1. Installation
2. Simple generic replicability case
3. Building containers
4. Using GPUs