USING SINGULARITY APPLICATION CONTAINERIZATION FOR REPRODUCIBLE SCIENTIFIC COMPUTING

VU HPC Course
23 Nov 2018

Maithili Kalamkar Stam
Raymond Oonk
Natalie Danezi
Lykle Voort

SURF SARA

Program

- 9.00 - 9.20 - Presentation: Introduction to containers
- 9.20 - 10.00 - Hands on: Interacting with Docker containers on your laptop
- 10:00 - 10.20 - Hands on: Installing Singularity on your Linux, Mac OSX or Windows laptop.
- 10.20 - 10.30 - Break
- 10.30 - 11:00 - Demo followed by Hands on: Build singularity images
- 11:00 - 12:30 - Hands on: Running Singularity containers on a Supercomputer

SURF SARA

The common problems you may have faced

- The software is a nightmare to install (missing libraries, dependencies)
- The software is incompatible with my OS (Windows vs Linux)
- The “I just need to test something” scenario
- The “it used to work before the update/upgrade” scenario
- My collaborator gave me a script that needs a different software version (Python 2 vs Python 3)
- Sharing workflows with your colleagues (not just a data analysis script but the software too)
- I do not have administrative (root) privileges on my machine/local cluster

How do you collaborate and share/distribute your work?

- Data
- Data analysis code

e.g., scp a tar file

- Software
- Software dependencies (libraries)

Set of instructions

and lots of luck!

How do you collaborate and share/distribute your work?

Distribute

Collaborate on HPC systems

Scale out

- Data
- Data analysis code
- Software
- Software dependencies (libraries)

What can containers do for you?

Isolate an application and its dependencies into a self-contained unit

Reduce dependency on underlying software, configuration and hardware

Provide full control of your environment, regardless of the host

- Data (small)
- Data analysis code
- Software
- Software dependencies (libraries)

Offer flexibility to your application

Reproducible research & collaboration: easy to distribute and validate your work

Mobility & computing agility in HPC

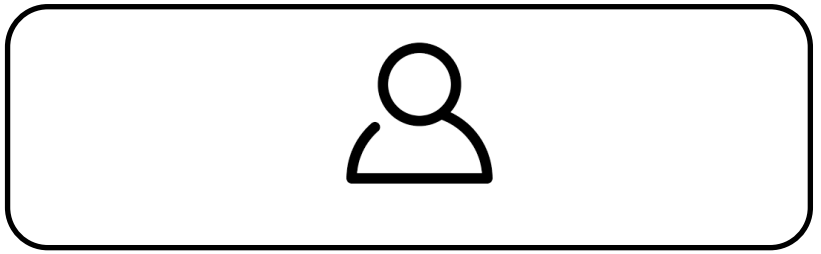
Operating system vs. Virtual Machine vs. Containers

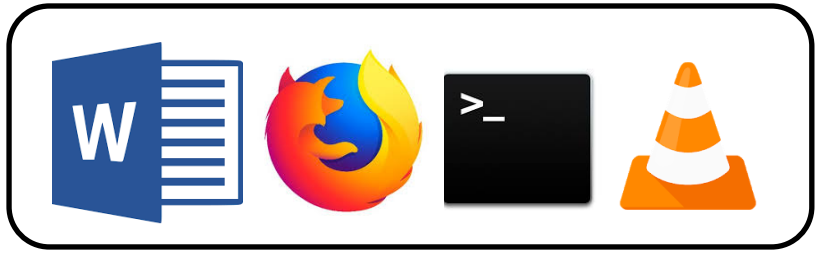
USER

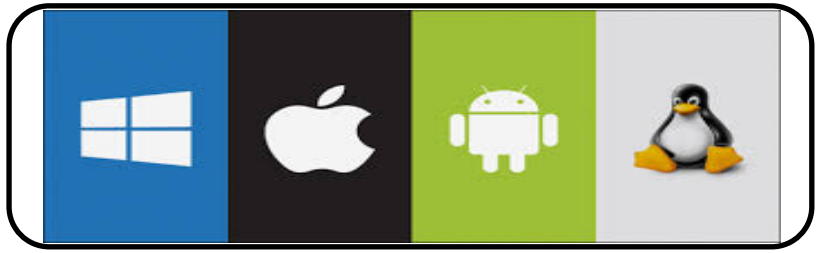
APPLICATIONS

OPERATING SYSTEM

HARDWARE

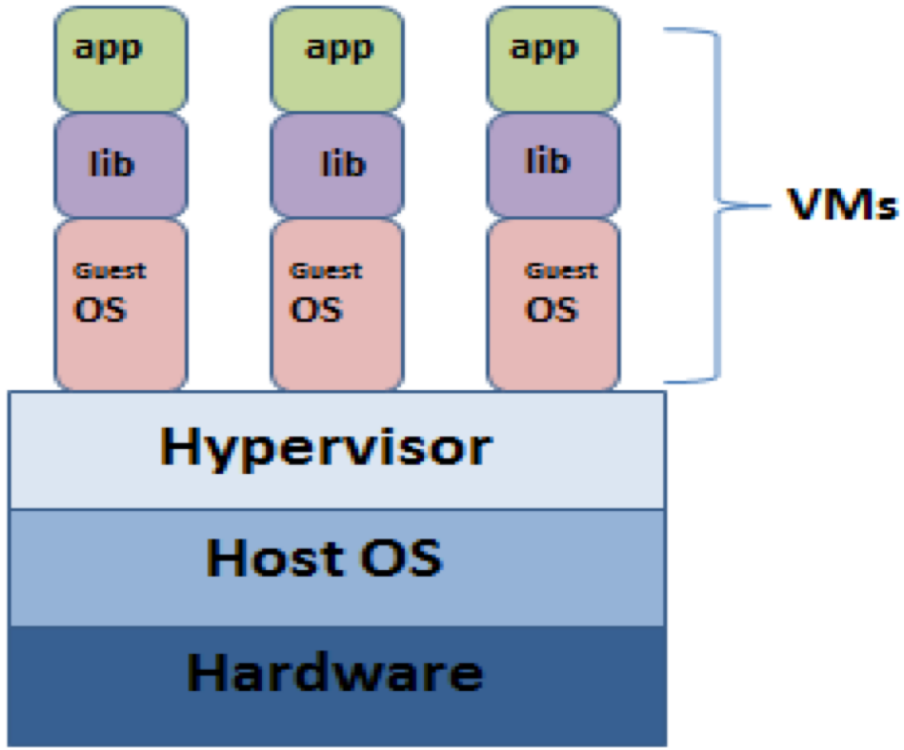






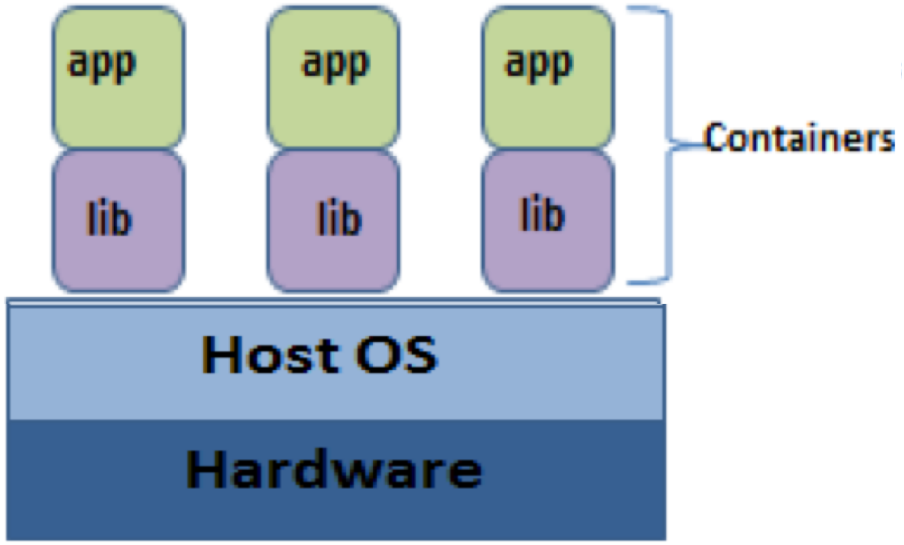
SURF SARA

Operating system vs. **Virtual Machine vs. Containers**



Virtual Machine Diagram

e.g., HPC Cloud at SURFsara



Container Diagram

e.g., Cartesius at SURFsara

Virtual Machine vs. Containers

Virtual Machines

- Have their own OS (e.g., Windows, Linux)
- Hardware virtualization (.e.g, CPU and RAM)
- Slow to boot /heavy
- More secure

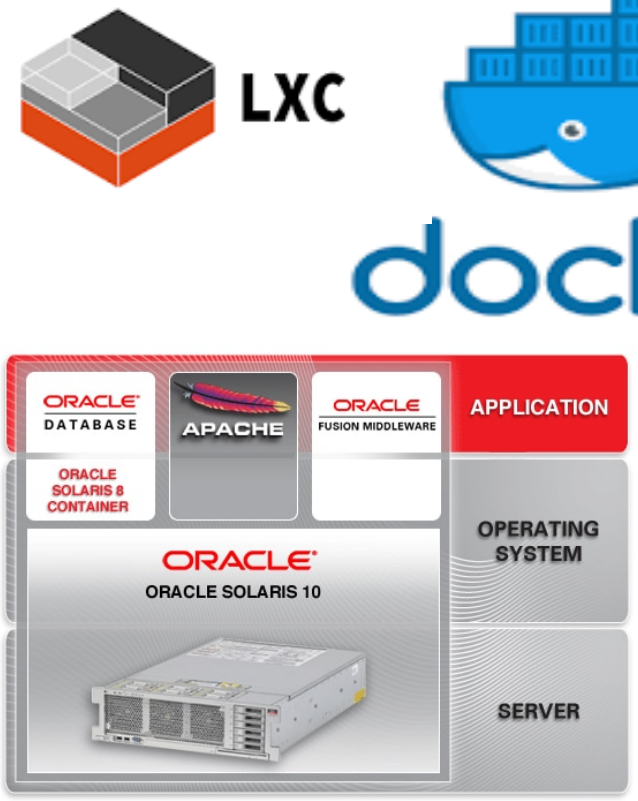
Containers

- Run on host OS and share its kernel (e.g., Linux)
- OS virtualization (decoupling applications from the OS)
- Faster performance / lightweight
- Less secure

Continuing with containers . . .

Container technologies and supporting tools (management and deployment)

This concept goes back to the 70's!



Singularity and Docker containers

- ✓ Open source projects
- ✓ Lightweight and fast
- ✓ Facilitate creation, maintenance and distribution of images
- ✓ Easy to install
- ✓ Compatibility with a variety of compute architectures

- ➢ Can be run by anyone – untrusted users running untrusted containers
- ➢ Native support for high-performance interconnects (e.g., Infiniband), GPUs, resource managers (e.g., SLURM)

- ➢ You need to be have admin privileges to run the containers
- ➢ Need some modifications to the software

SURF SARA

Continuing to our hands on session

Requirements:

1. Account on Cartesius (e.g., sdemo000) – if you don't have one, please tell us now
2. Docker installed on your laptop – If you did not manage to do it, pair up with someone who has it on their laptop
3. Instructions for hands-on session https://github.com/sara-nl/singularity-course

SURF SARA