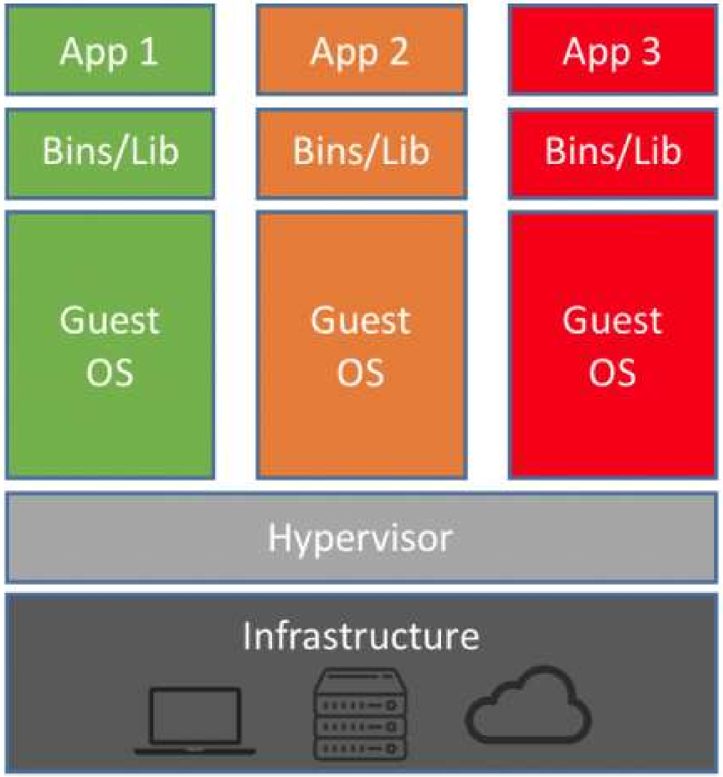
Introduction to Singularity: Creating and Running Containers on HPC

Le Yan

Virtualization

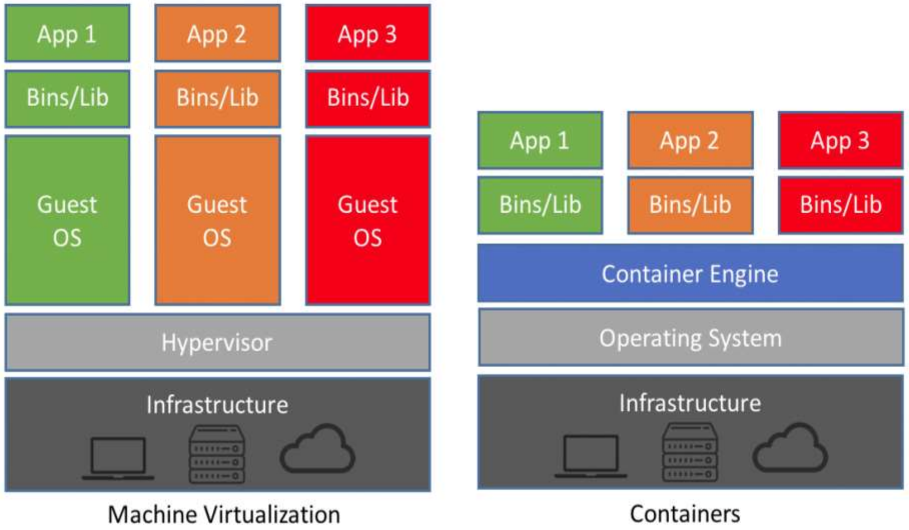
- Virtualization allows users to run multiple operating system instances on the same server
- Multiple applications can share the hardware resources on the server



https://blog.netapp.com/blogs/containers-vs-vms/

Virtual Machines vs. Containers

Virtual machines	Containers
Very flexible, e.g. one can run a Windows guest OS on Linux or vice versa	Less flexible, On Linux systems only
Heavyweight, need to install all files of a guest OS	Very lightweight, will use the kernel of the host OS



https://blog.netapp.com/blogs/containers-vs-vms/

What is Singularity

- Singularity is a open-source container software that allows users to pack an application and all of its dependencies into a single image (file)
- Developed by Greg Kurtzer at Lawrence Livermore National Laboratory
- “Container for HPC”
- Native command line interface
  - Syntax: `singularity <command> <options> <arguments>`

Containers: Docker vs. Singularity

Docker	Singularity
Assumes that the user has root privilege in the production environment	Does not assume that the user has root privilege in the production environment
Mature	Less mature, very active development
Designed for system services	Designed for HPC use cases

Why Singularity: HPC Users' Perspective

Pain points using HPC
Dependencies of an application are available on the host OS, e.g. GLIBC version too low
Dependencies of an application are complex and difficult to resolve/install
Reproducibility is not guaranteed
Difficult to share workflows, pipelines and environments

Why Singularity: HPC Users' Perspective

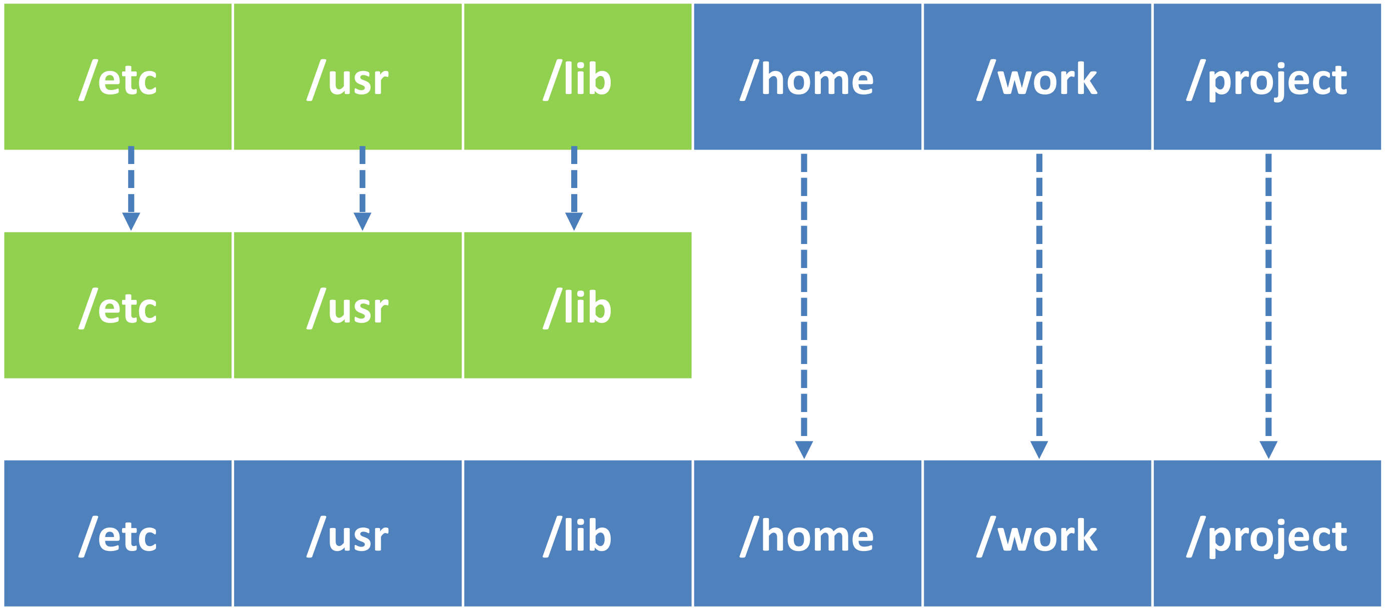
Pain points using HPC	With Singularity
Dependencies of an application are available on the host OS, e.g. GLIBC version too low	Build an image with an different OS
Dependencies of an application are complex and difficult to resolve/install	Obtain an image (from the developer or other users)
Reproducibility is not guaranteed	Build and share an image
Difficult to share workflows, pipelines and environments	Build and share an image

Singularity on QB2

- Singularity is installed on all compute nodes
- Still in “friendly user” mode, which means
  - Users can only run images from a specific directory
    - Located under `/home/admin/singularity`
  - The images are built by HPC staff
    - Users can request Singularity images
- It will be in “production” so users can build and upload their own images

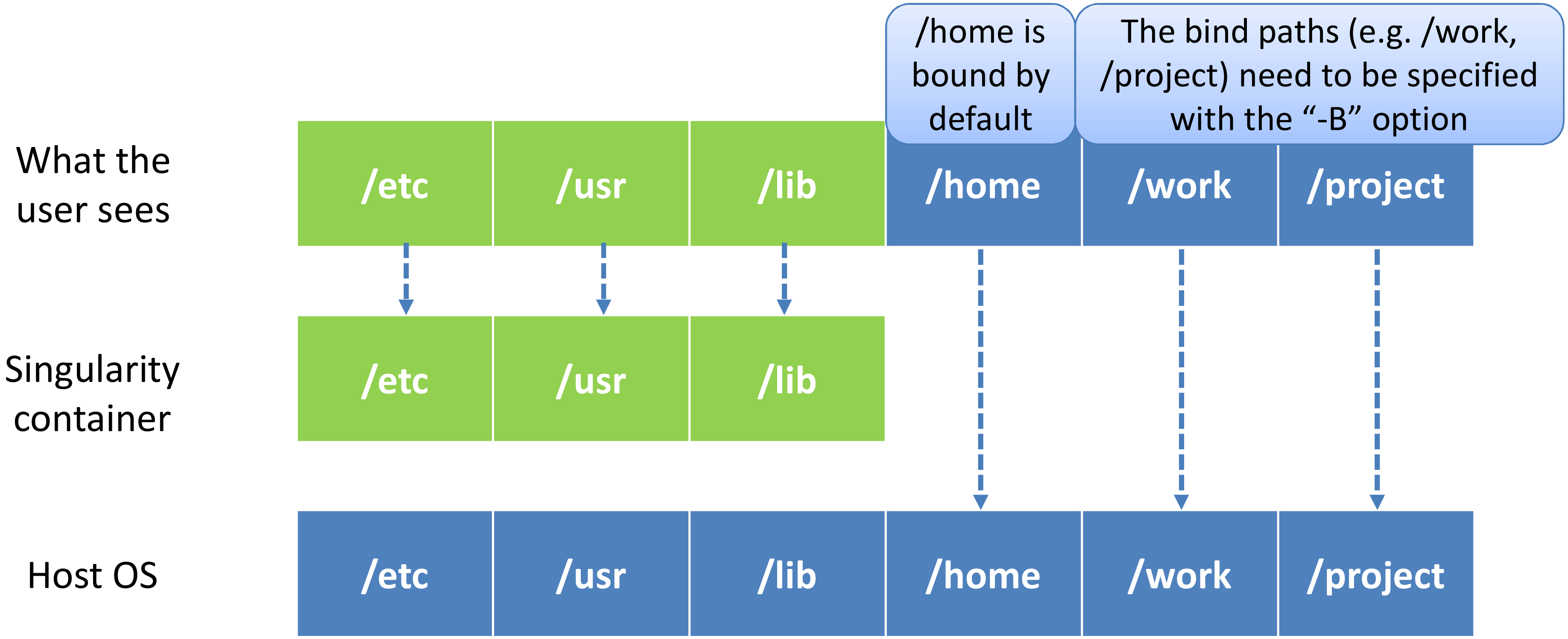
Demo

Overlay File System



What the user sees	/etc	/usr	/lib	/home	/work	/project
Singularity container	/etc	/usr	/lib			
Host OS	/etc	/usr	/lib	/home	/work	/project

Overlay File System



Privilege Escalation

- A very important feature of Singularity: If you don’t have root privileges outside the container, you won’t be able to obtain root privileges inside the container.

Singularity Workflow

- Step 1: Install Singularity on a local Linux machine (or VM)
  - Root privilege is needed
- Step 2: Build Singularity images on the local machine
  - Root privilege is needed
- Step 3: Upload images onto the HPC cluster
  - Root privilege is NOT needed
- Step 4: Run images on the HPC cluster
  - Root privilege is NOT needed

Installing Singularity

- On Linux
  - Install binary (recommended)
    - Use either `apt-get` or `rpm/yum`
  - Install from source
    - https://github.com/sylabs/singularity
- On Windows or Mac
  - Install a Linux VM first

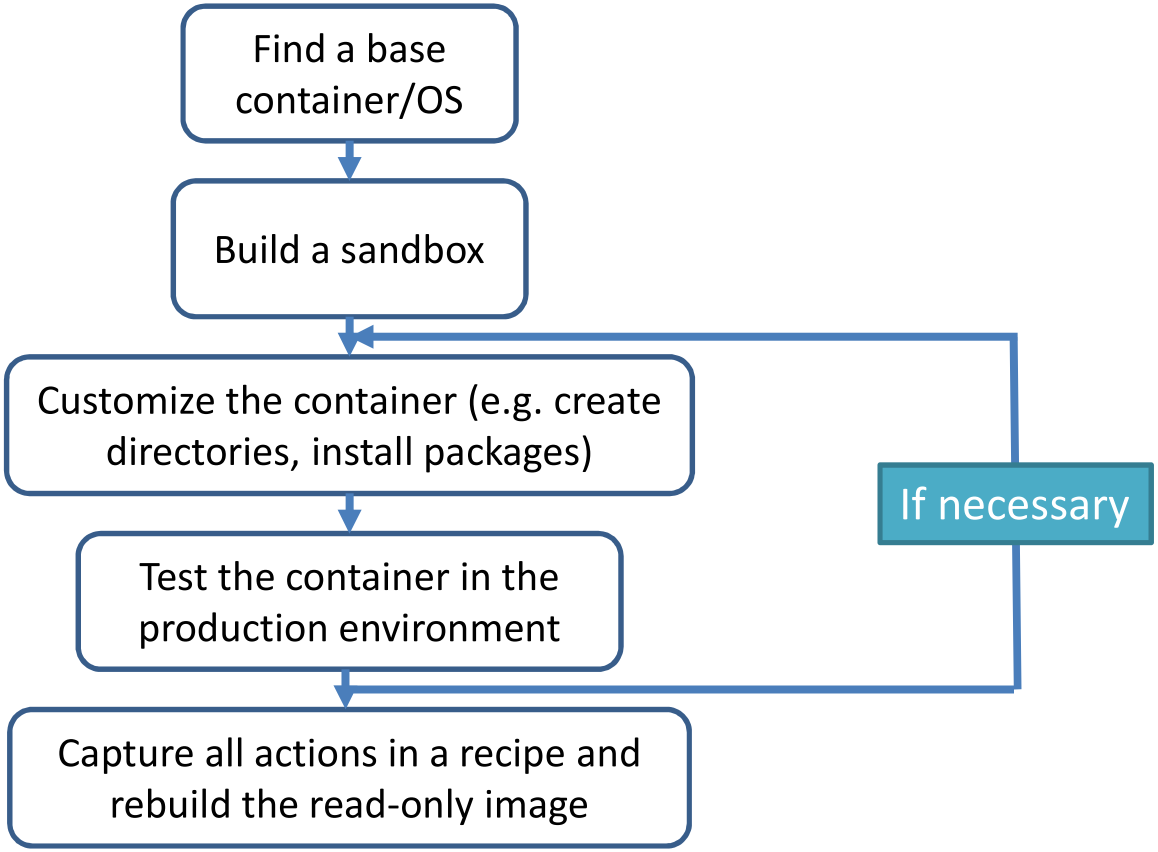
Building Singularity Images

- Use the "`build`" command to build Singularity images
- Syntax:
  - `singularity build [build options…] <container path> <BUILD TARGET>`
- The "BUILD TARGET" defines the method how an image is built
  - A URI to a base OS/container image
  - Path to a Singularity sandbox
  - Path to a Singularity recipe (definition file)
- Build options
  - By default, a compressed, read-only image will be built
  - The "`--writable`" option builds a writable image
  - The "`--sandbox`" option builds a sandbox

Where to Find Base OS/Containers

- Docker hub: https://hub.docker.com
- Singularity hub: https://singularity-hub.org
- NVIDIA GPU Cloud: https://ngc.nvidia.com
- Distribution repos
  - YUM/RHEL
  - Debian/Ubuntu

Building Singularity Images



Find a base container/OS

Build a sandbox

Customize the container (e.g. create directories, install packages)

Test the container in the production environment

Capture all actions in a recipe and rebuild the read-only image

Demo

Singularity Definition Files (Recipes)

- Capture all the interactive building steps

```
BootStrap: docker
From: ubuntu:16.04

%labels
    Author lyan1@lsu.edu

%post
apt-get update && apt-get install -y vim

# Create bind points for HPC environment
mkdir /project /work

%environment
export LC_ALL=C

%runscript
echo "Hello, world!"
```

Header: describes the base container image

Label: metadata for the container

Post: commands executed within the container after the base OS has been installed at build time.

Environment: define environment variables

Runscript: commands that will be run when the container is run by “singularity run”

Inspecting Singularity Images

- Use the “inspect” command to query
  - How an image is built
  - What the runscript is
  - What environment variables are set
- Syntax: `singularity inspect [options] <container image>`
  - The options are self-explanatory: “--labels”, “--runscript”, “--deffile”, “--environment” etc.

Running Singularity on QB2

- Syntax
  - `Singularity <command> [options] <container image>`
- Commands
  - `shell`: run an interactive bash shell within container
  - `run`: launch a runscript within container
  - `exec`: execute a command within container

Running Singularity on QB2

- Singularity can be embedded in a job script just like any other application

```
#!/bin/sh
#PBS -A your_allocation
#PBS -q bigmem
#PBS -l nodes=1:ppn=48
#PBS -l walltime=24:00:00
#PBS -N Cactus_Singularity

Cd PBS_O_WORKDIR
singularity exec -B /work /home/admin/singularity/cactus-ubuntu-
16.04.simg cactus --binariesMode local --maxMemory 100G
/work/lyan1/clustertest/cactus/jobstore evolverMammals.txt
/work/lyan1/clustertest/cactus/output
```

Demo

Questions?