Magic Tools to Install & Manage Software

Part 2: Singularity Container

Jason Li
HPC User Services
LSU HPC / LONI
sys-help@loni.org

Louisiana State University, Baton Rouge
Mar 20, 2024

Magic Tools to Install & Manage Software

Part 1: CONDA Virtual Environment

Part 2: Singularity Container

Outlines

1. Why Container?
2. Run an Existing Container Image
3. Get More Container Images
4. Build Your Own Container Image

Outlines

1. **Why Container?**
   1) Problems
   2) Container & Singularity

2. **Run an Existing Container Image**
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. **Get More Container Images**
   1) What you need
   2) Where to get
   3) How to get

4. **Build Your Own Container Image**
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

Outlines

1. **Why Container?**
   1) Problems
   2) Container & Singularity
2. **Run an Existing Container Image**
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity
3. **Get More Container Images**
   1) What you need
   2) Where to get
   3) How to get
4. **Build Your Own Container Image**
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

Outlines

1. **Why Container?**
   1) Problems
   2) Container & Singularity
2. **Run an Existing Container Image**
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity
3. **Get More Container Images**
   1) What you need
   2) Where to get
   3) How to get
4. **Build Your Own Container Image**
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

1) Problems

- **Core problem:**

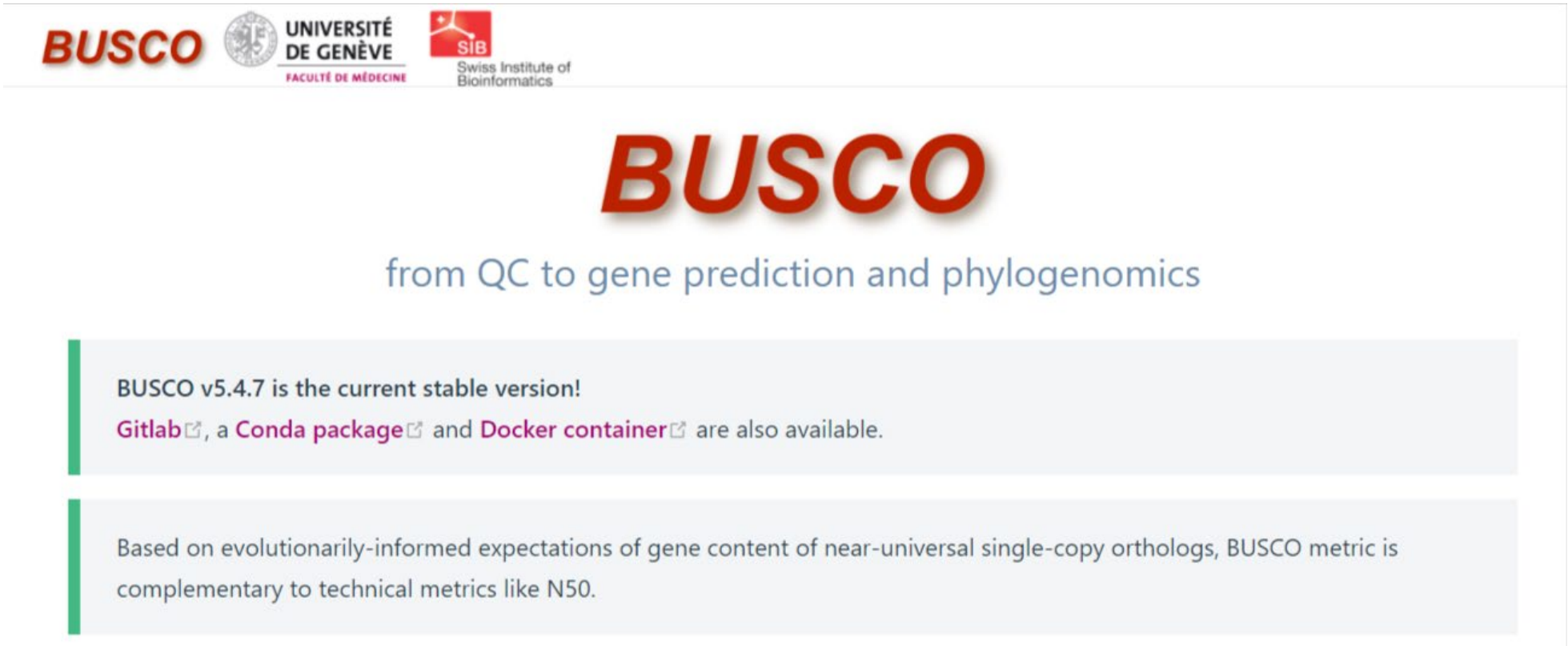
Installing software on an HPC system

1) Problems

- **Traditional Linux solution:**
  - Compiling from source code

1) Problems

a) Dependencies (Welcome to Linux!)



1) Problems

a) **Dependencies** (Welcome to Linux!)

Third-party components

A full installation of BUSCO requires *Python 3.3+* (2.7 is not supported from v4 onwards), *BioPython*, *pandas*, *BBMap*, *tBLASTn 2.2+*, *Augustus 3.2+*, *Prodigal*, *Metaeuk*, *HMMER3.1+*, *SEPP*, and *R + ggplot2* for the plotting companion script. Some of these tools are necessary only for analysing certain type of organisms and input data, or for specific run modes.

- https://biopython.org/
- https://pandas.pydata.org/
- https://jgi.doe.gov/data-and-tools/software-tools/bbtools/
- https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST
- http://bioinf.uni-greifswald.de/augustus/
- https://github.com/soedinglab/metaeuk
- https://github.com/hyattpd/Prodigal
- http://hmmer.org/
- https://github.com/smirarab/sepp/
- https://www.r-project.org/

> Please make sure that each software package listed above works INDEPENDENTLY of BUSCO before attempting to run any BUSCO assessments.

1) Problems

a) **Dependencies** (Welcome to Linux!)

Third-party components

A full installation of BUSCO requires *Python 3.3+* (2.7 is not supported from v4 onwards), *BioPython*, *pandas*, *BBMap*, *tBLASTn 2.2+*, *Augustus 3.2+*, *Prodigal*, *Metaeuk*, *HMMER3.1+*, *SEPP*, and *R + ggplot2* for the plotting companion script. Some of these tools are necessary only for analysing certain type of organisms and input data, or for specific run modes.

- https://biopython.org/
- https://pandas.pydata.org/
- https://jgi.doe.gov/data-and-tools/software-tools/bbtools/
- https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST
- http://bioinf.uni-greifswald.de/augustus/
- https://github.com/soedinglab/metaeuk
- https://github.com/hyattpd/Prodigal
- http://hmmer.org/
- https://github.com/smirarab/sepp/
- https://www.r-project.org/

> Please make sure that each software package listed above works INDEPENDENTLY of BUSCO before attempting to run any BUSCO assessments.

1) Problems

a) **Dependencies** (Welcome to Linux!)

Third-party components

A full installation of BUSCO requires *Python 3.3+* (2.7 is not supported from v4 onwards), *BioPytho*
pandas, *BBMap*, *tBLASTn 2.2+*, *Augustus 3.2+*, *Prodigal*, *Metaeuk*, *HMMER3.1+*, *SEPP*, and *R + ggplo*
the plotting companion script. Some of these tools are necessary only for analysing certain type o
organisms and input data, or for specific run modes.

- https://biopython.org/
- https://pandas.pydata.org/
- https://jgi.doe.gov/data-and-tools/software-tools/bbtools/
- https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST
- http://bioinf.uni-greifswald.de/augustus/
- https://github.com/soedinglab/metaeuk
- https://github.com/hyattpd/Prodigal
- http://hmmer.org/
- https://github.com/smirarab/sepp/
- https://www.r-project.org/

> Please make sure that each software package listed above works INDEPENDENTLY of BUSCO
> before attempting to run any BUSCO assessments.

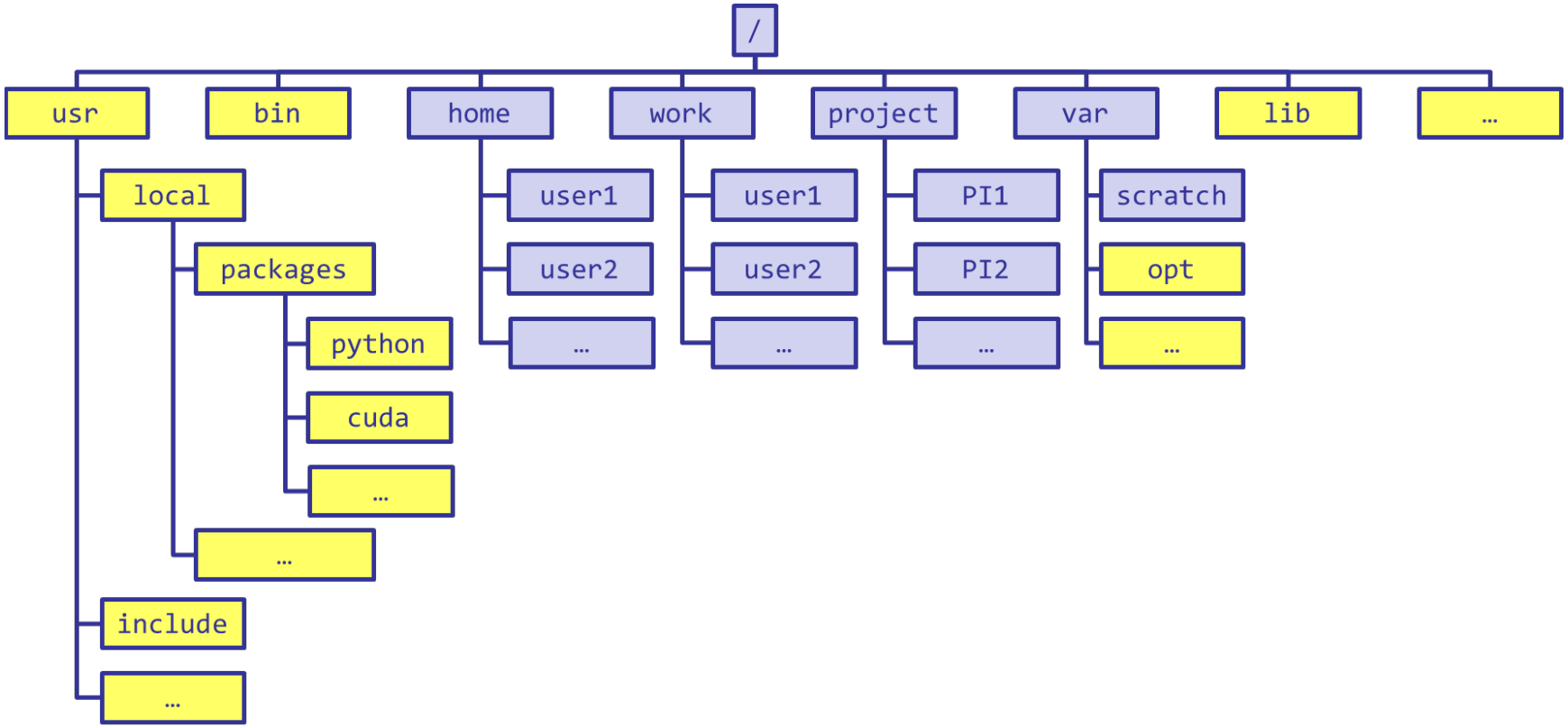
- Dependencies

The following dependencies are required for AUGUSTUS:

- for gzip compressed input: (set ZIPINPUT = false in common.mk if t… re not available)
  - libboost-iostreams-dev
  - zlib1g-dev
- for comparative AUGUSTUS (multi-species, CGP): (set COMPGENEPRED = false in co… raries required by the CGP version are not available. Augustus can then only be run in single-genome mode, which is what most users need.)
  - libgsl-dev
  - libboost-all-dev
  - libsuitesparse-dev
  - liblpsolve55-dev
  - libsqlite3-dev (add SQLITE = false to common.mk if this feature is not required or the required library is not available)
  - libmysql++-dev (add MYSQL = false to common.mk if this feature is not required or the required library is not available)
- for compiling utilities bam2hints and filterBam:
  - libbamtools-dev zlib1g-dev
- for compiling utility utrrnaseq:
  - libboost-all-dev (version must be >Boost_1_49_0)
- for compiling utility bam2wig:
  - Follow these instructions. Note that it shouldn't be a problem to compile AUGUSTUS without bam2wig. In practice, you can simply use `bamToWig.py` to accomplish the same task.
- For compiling homgenemapping (set BOOST = FALSE in auxprogs/homgenemapping/src/Makefile if the option --printHomologs is not required or the required libraries are not available)
  - libboost-all-dev
- for scripts:
  - Perl
  - Python3
- for the python3 script bamToWig.py:
  - twoBitInfo and faToTwoBit from http://hgdownload.soe.ucsc.edu/admin/exe . bamToWig.py will automatically download these tools to the working directory during execution if they are not in your $PATH.
  - SAMtools (available e.g. via package managers or here - see notes below)

1) Problems

b) Permission denied (Welcome to HPC!)



1) Problems

b) Permission denied (Welcome to HPC!)

```
[jasonli3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ module li
Currently Loaded Modulefiles:
 1) python/3.6.2-anaconda-tensorflow
```

1) Problems

b) Permission denied (Welcome to HPC!)

```
[jasonli3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ module li
Currently Loaded Modulefiles:
 1) python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ pip install geos
Collecting geos
  Downloading https://files.pythonhosted.org/packages/49/5b/b8acf74c01187a36aa41b6523deb9baa59c
    100% |                                | 409kB 3.0MB/s
```

1) Problems

b) Permission denied (Welcome to HPC!)

```
[jasonli3@smic2 ~]$ module load python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ module li
Currently Loaded Modulefiles:
 1) python/3.6.2-anaconda-tensorflow
[jasonli3@smic2 ~]$ pip install geos
Collecting geos
  Downloading https://files.pythonhosted.org/packages/49/5b/b8acf74c01187a36aa41b6523deb9baa59c
    100% |                                | 409kB 3.0MB/s
```

```
    os.makedirs(path)
  File "/usr/local/packages/python/3.6.2-anaconda/lib/python3.6/os.py", line 220, in makedirs
    mkdir(name, mode)
PermissionError: [Errno 13] Permission denied: '/usr/local/packages/python/3.6.2-anaconda/lib/p
You are using pip version 9.0.1, however version 23.0.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
[jasonli3@smic2 ~]$
```

1) Problems

b) Permission denied (Welcome to HPC!)

- If you ask Google / ChatGPT…

```
$ sudo yum install …

$ sudo apt-get install …

$ sudo make install
```

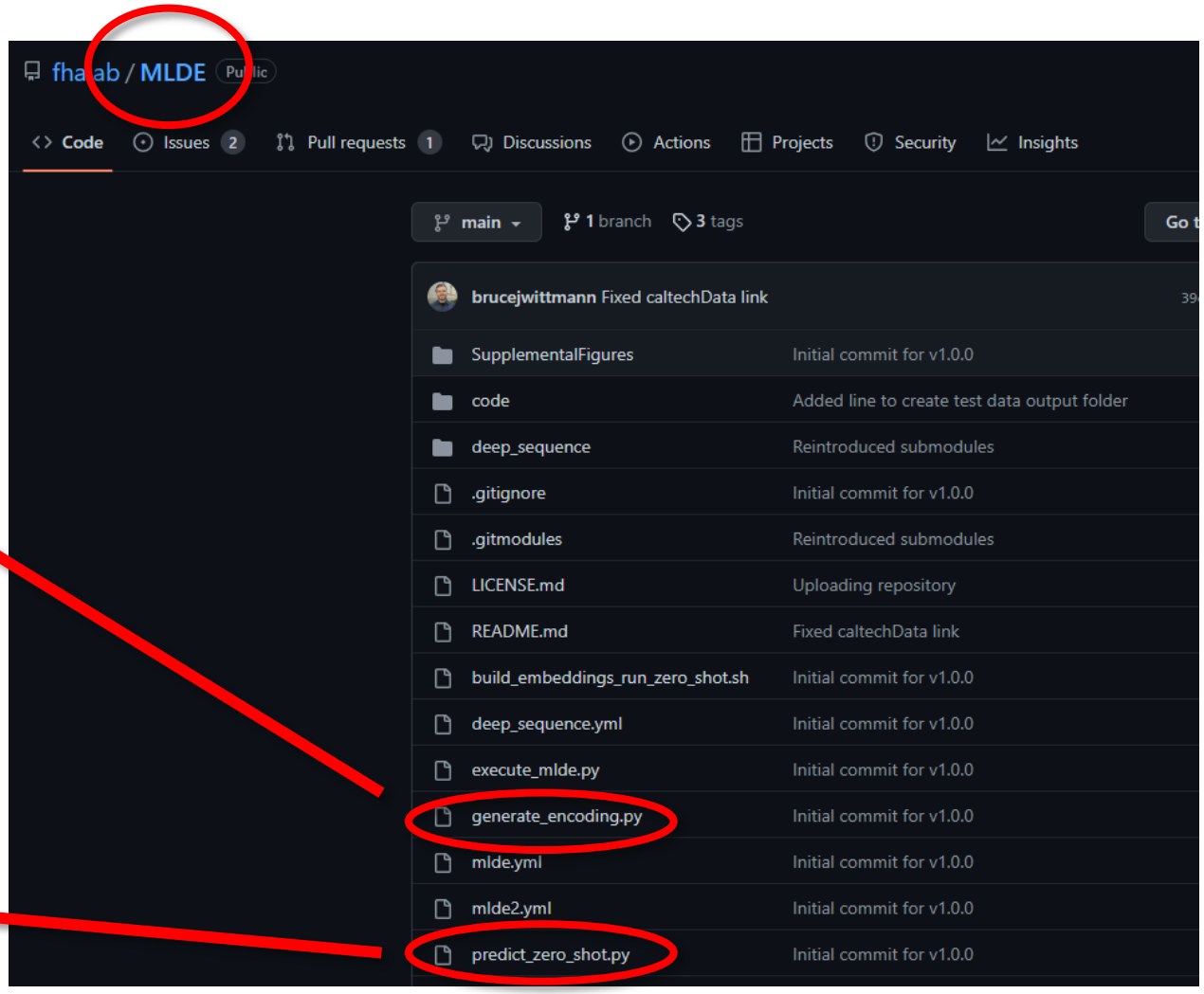
1) Problems

c) Conflicted packages

- What if I need two packages w/ conflicted dependencies?

Tensorflow 1.13

PyTorch > 1.5



d) Sharing / Migrating your environment

– Huge effort & large disk quota to install

- What if my colleagues want to use?
- What if I want to migrate a different cluster?

1) Problems

Any of those apply to you?

Magic Tools to Install / Manage Software

Part 1: CONDA Virtual Environment

Part 2: Singularity Container

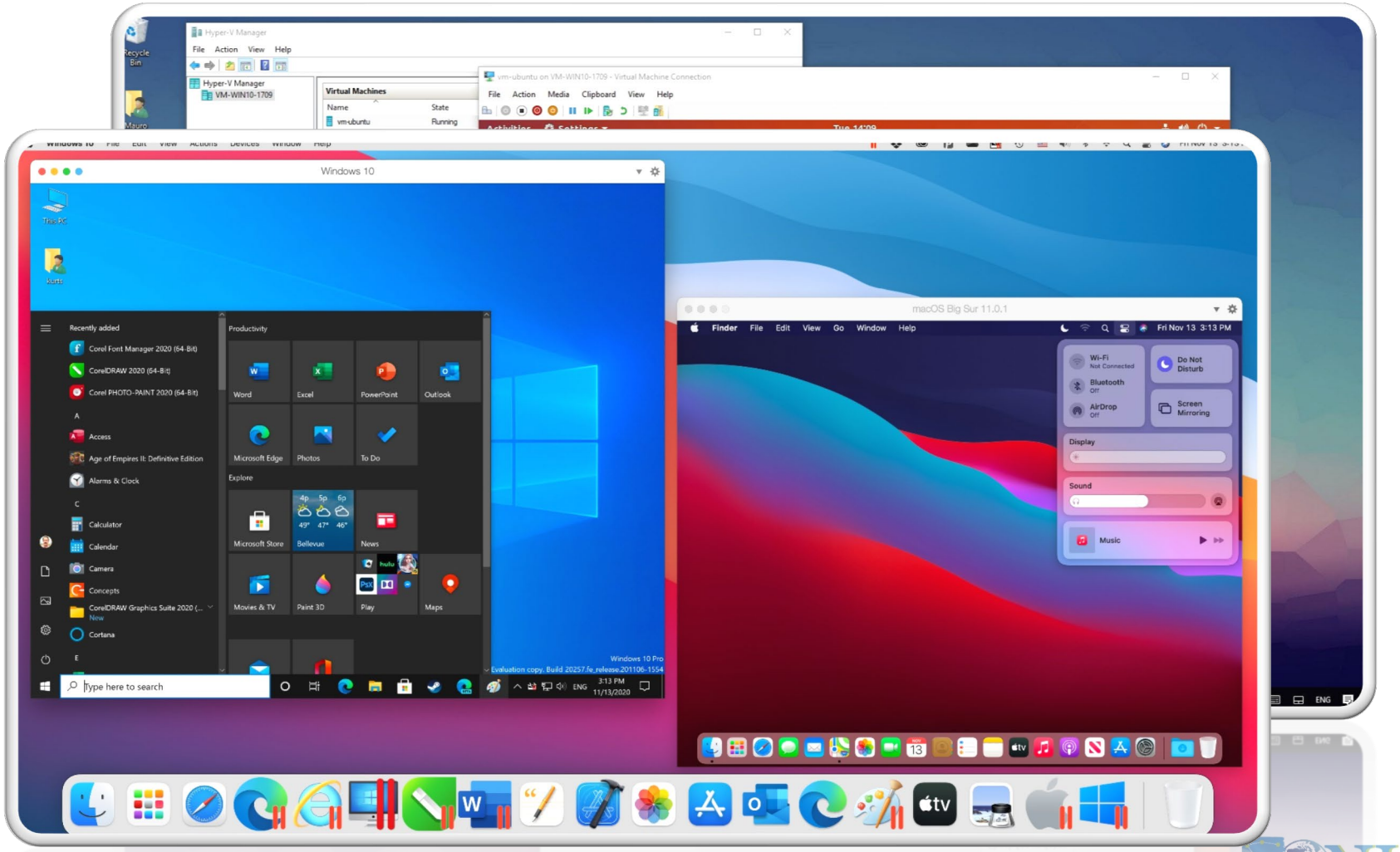
Outlines

1. **Why Container?**
   1) Problems
   2) Container & Singularity
2. **Run an Existing Container Image**
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity
3. **Get More Container Images**
   1) What you need
   2) Where to get
   3) How to get
4. **Build Your Own Container Image**
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

2) Container & Singularity

a) What is a container?

- **Virtual machine**
  - "Virtualize" / "mimic" an **entire computer** on another computer
  - Virtualize both **hardware** and **software**



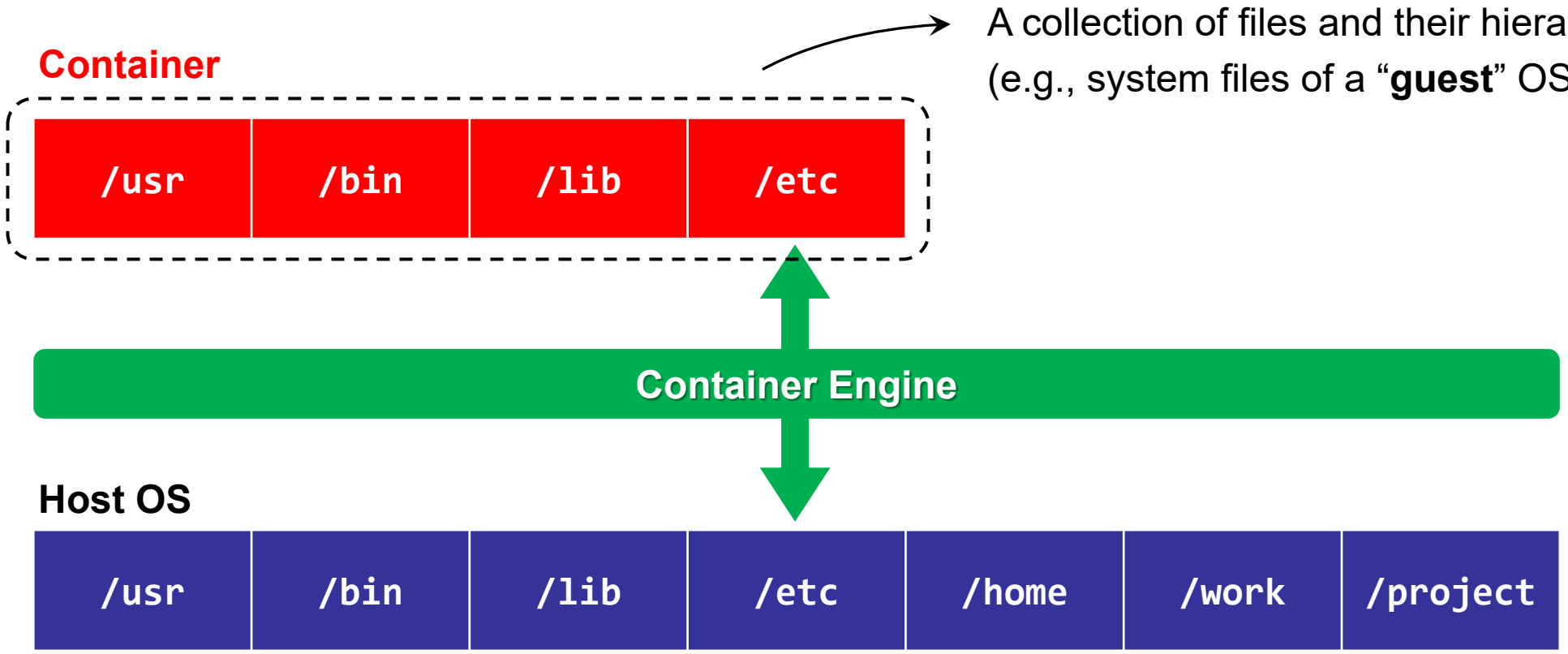
LSU INFORMATION TECHNOLOGY SERVICES

a) What is a container?

- **Container:**
  - A **lightweight** and **fast** virtual machine
  - Only virtualize the **Operation System** (meaning, does not virtualize hardware)
  - Only virtualize **Linux** on **Linux**

2) Container & Singularity

a) What is a container?



2) Container & Singularity

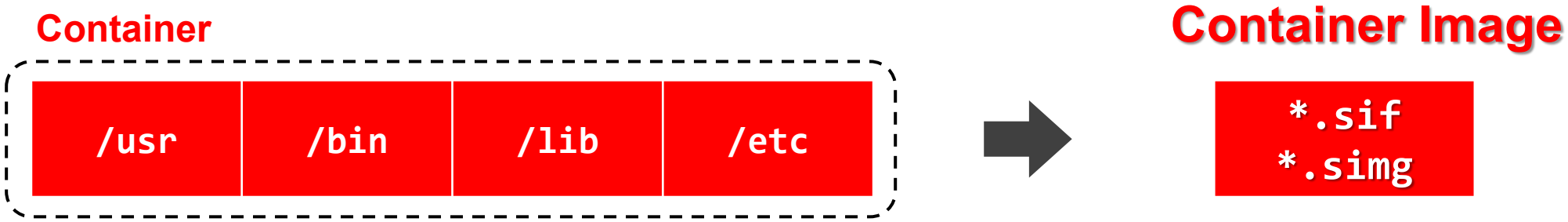
a) What is a container?

/usr	/bin	/lib	/etc	/home	/work	/project

- **A "chimera" system:**
  - Can virtualize **an entirely different OS** !
  - Can contain other **software packages** (inc. dependencies, environment settings, etc.) installed in the guest OS

2) Container & Singularity

a) What is a container?



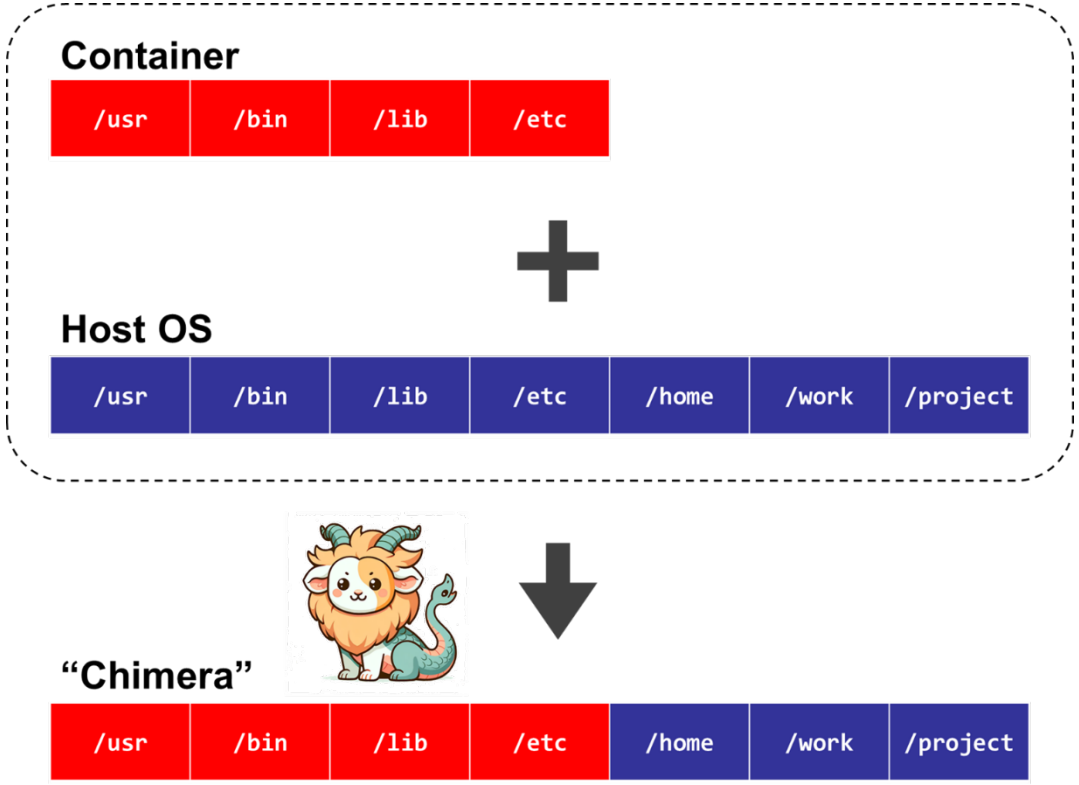
2) Container & Singularity

a) What is a container?

- **Properties:**
  - **Self-contained**

    All dependencies can be installed within the container
  - **Isolated**

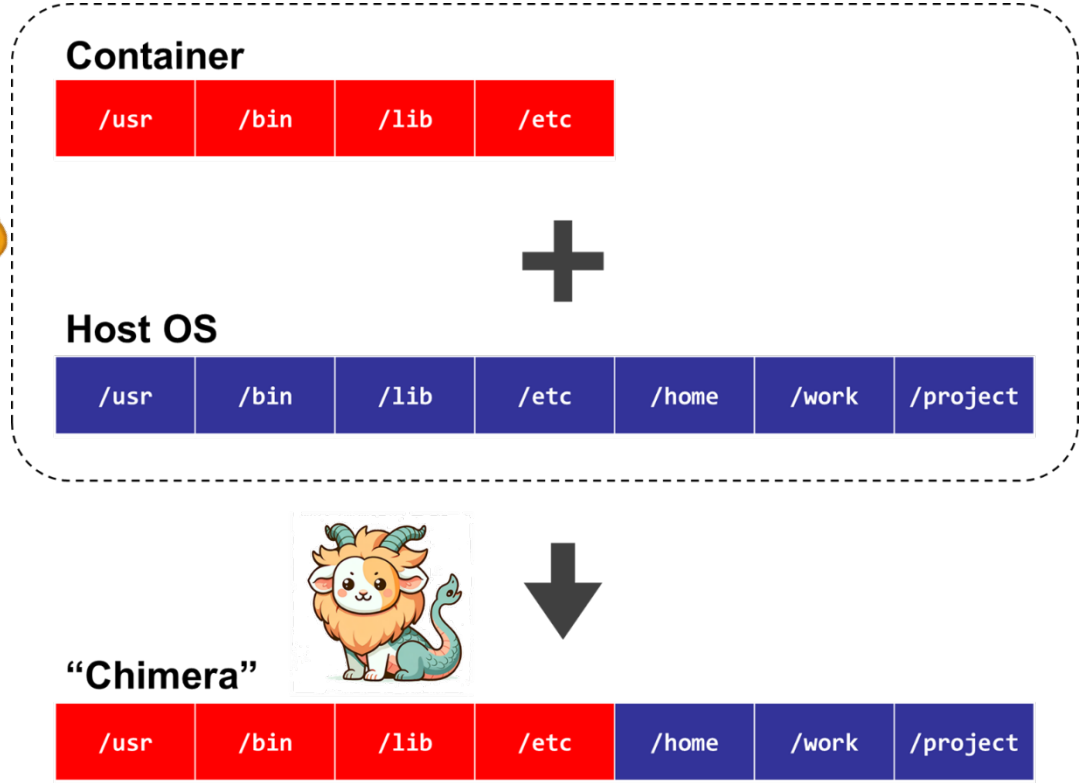
    Whatever happens in a container stays in that container…



b) How does it solve my problems?

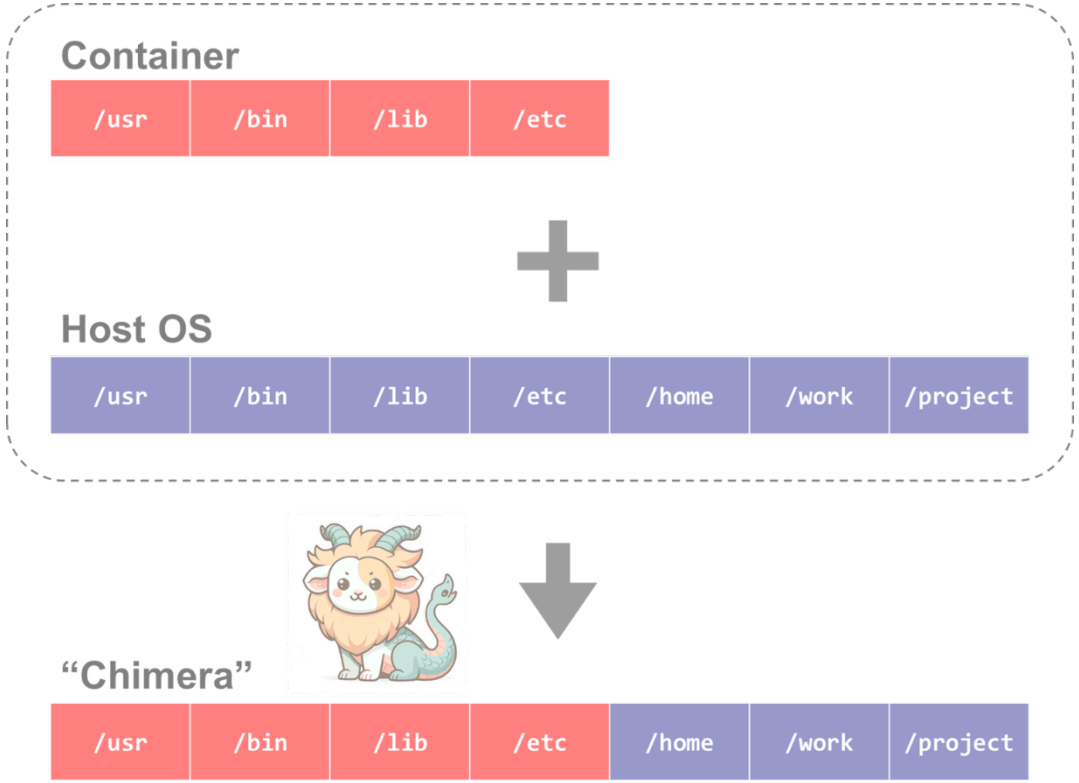
- **Dependency issue**
  - Pack all dependencies (even OS) in container
  - Can use `apt-get` or `yum`
  - **Developers now release containers!**

- **Permission issue**
  - Can't write to certain paths on HPC, but **CAN** write to them in container
- **Conflicted packages**
  - Install in different containers.
- **Share / Migrate**
  - Copy-paste a container image!



c) What is Singularity?

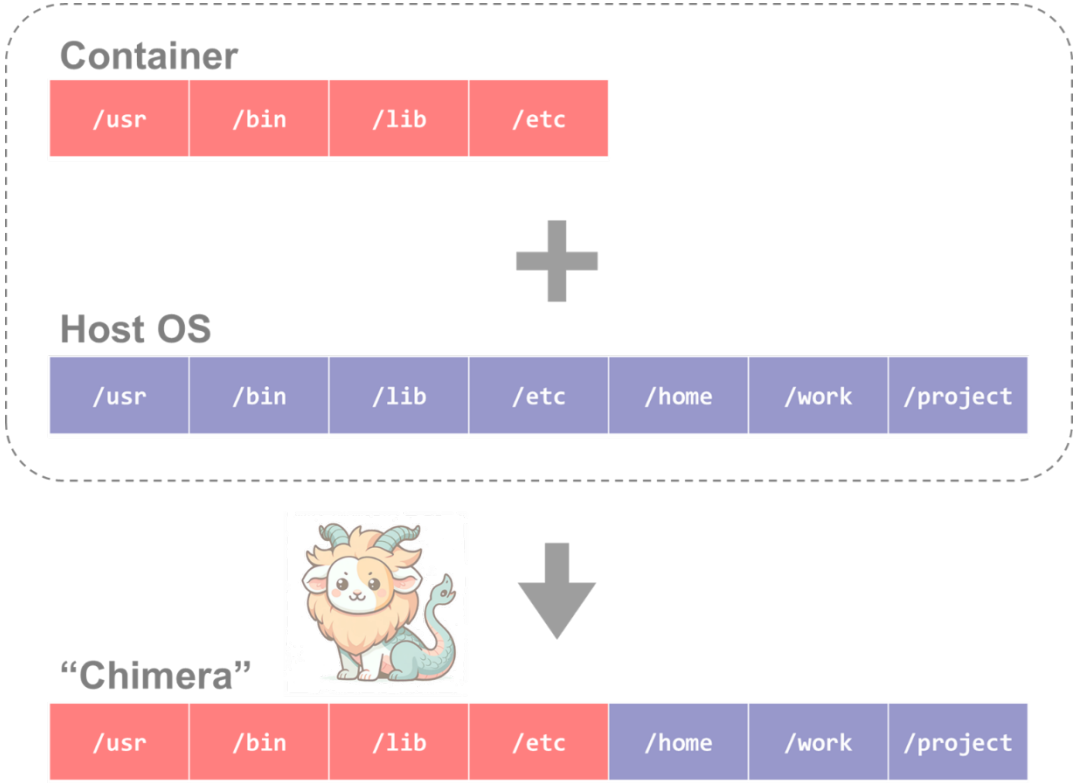
Technology →



c) What is Singularity?

Singularity

↑ **Software** system that implements the technology



c) What is Singularity?

c) What is Singularity?

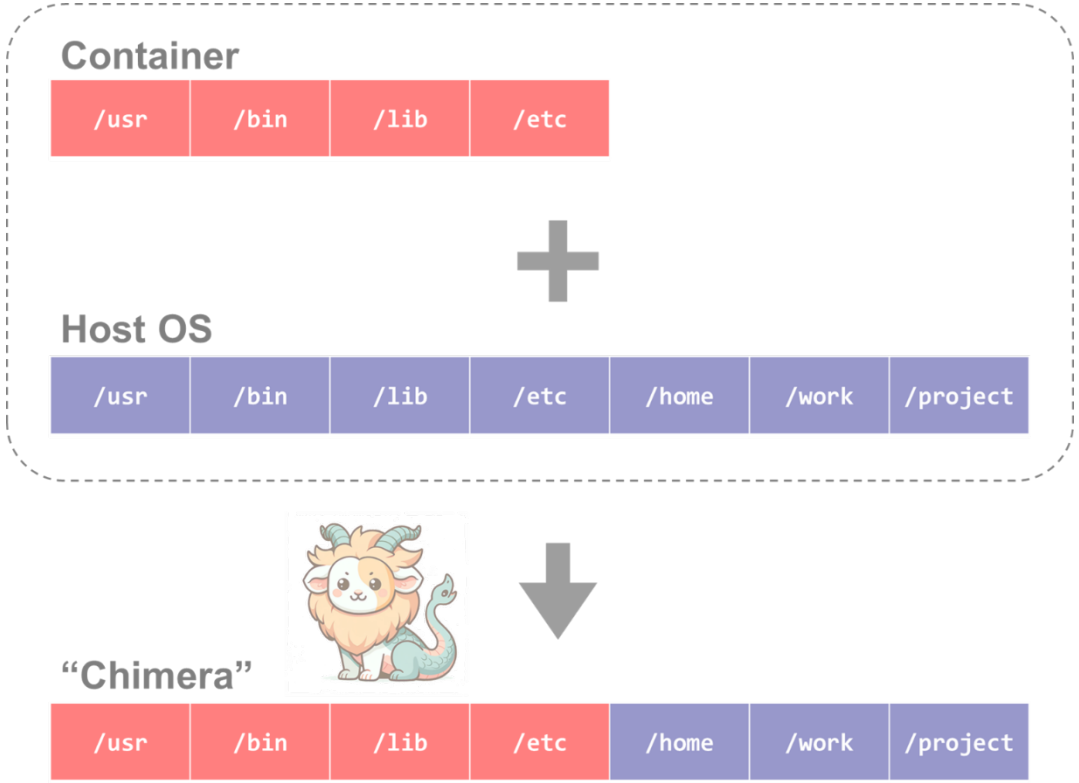
- Does **NOT** need root privileges
- "**Container for HPC**"

- **Needs** root privileges

Summary

Technology that helps with software installation →

↓ **Software** system that implements the technology



Outlines

1. Why Container?
   1) Problems
   2) Container & Singularity
2. **Run an Existing Container Image**
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity
3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get
4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

Outlines

1. Why Container?
   1) Problems
   2) Container & Singularity
2. **Run an Existing Container Image**
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity
3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get
4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

1) What you need

- **Singularity availability**

  a) On **all clusters**
  - ✓ **LSU HPC**: SMIC, Deep Bayou, SuperMike 3
  - ✓ **LONI**: QB2, QB3

  b) Only on **computing nodes**
  - × Unavailable on head nodes
  - ✓ Must start a job (interactive & batch)

  c) To **all users**
  - × No additional action required

1) What you need

To …	What you need …
Run an Existing Container Image	• Access to our HPC systems - An active account - An active allocation

1) What you need

Outlines

1. Why Container?
   1) Problems
   2) Container & Singularity
2. **Run an Existing Container Image**
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity
3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get
4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

LSU INFORMATION TECHNOLOGY SERVICES

2) Basic commands

- **Available images**

  – On all clusters: `/home/admin/singularity`

```
(base) [jasonli3@mike1 ~]$ ls /home/admin/singularity
abinit.v9.8.3.sif                   jax-0.4.13-gpu-jupyter.sif
alphafold-catgumag-2.2.sif          MuST-1.8.7.sif
bowtie2-2.5.1.sif                   openpose.sif
busco-5.4.7.sif                     pytorch-2.0.1-gpu-jupyter.sif
clara.400-1.sif                     rstudio-2023.03.1-446-rocky8.sif
deepmd-kit_2.0.3_cuda11.3_gpu.sif   salmon-1.10.2.sif
deepmd-kit_2.2.1_cuda11.6_gpu.sif   tensorflow-2.13.0-gpu-jupyter.sif
delft3dfm_r142632_05032023.sif      trinity-2.15.1.sif
delft3dfm_r142632.sif               ubuntu-training.sif
delft3d_r142586.sif
```

2) Basic commands

a) Common usage 1: Open a shell in the image

Syntax		Description
`singularity shell`	`<container>`	Starts a shell in the image

Try me: `/home/admin/singularity/ubuntu-training.sif`

2) Basic commands

a) Common usage 1: Open a shell in the image

Syntax		Description
`singularity shell [options] <container>`		Starts a shell in the image
`[Options]`	`-B /path/to/bind`	Bind a path(s) • /home is bound by default
	`--nv`	Enable Nvidia GPU

2) Basic commands

b) Common usage 2: Execute a single command in the image

Syntax	Description
`singularity exec <container> <command>`	Execute a command in the image

Try me: `/home/admin/singularity/ubuntu-training.sif`

2) Basic commands

b) Common usage 2: Execute a single command in the image

Syntax		Description
`singularity exec [options] <container> <command>`		Execute a command in the image
`[Options]`	`-B /path/to/bind`	Bind a path(s) • /home is bound by default
	`--nv`	Enable Nvidia GPU

c) Another (less) common usage: Run a prewritten script

Syntax		Description
`singularity run [options] <container>`		Run a prewritten script
`[Options]`	`-B /path/to/bind`	Bind a path(s) • /home is bound by default
	`--nv`	Enable Nvidia GPU

2) Basic commands

- **Quick recap**

Syntax	Description
`singularity shell [options] <container>`	Run a prewritten script
`singularity exec [options] <container> <command>`	Execute a command in the image
`singularity run [options] <container>`	Run a prewritten script

Outlines

1. Why Container?
   1) Problems
   2) Container & Singularity

2. **Run an Existing Container Image**
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

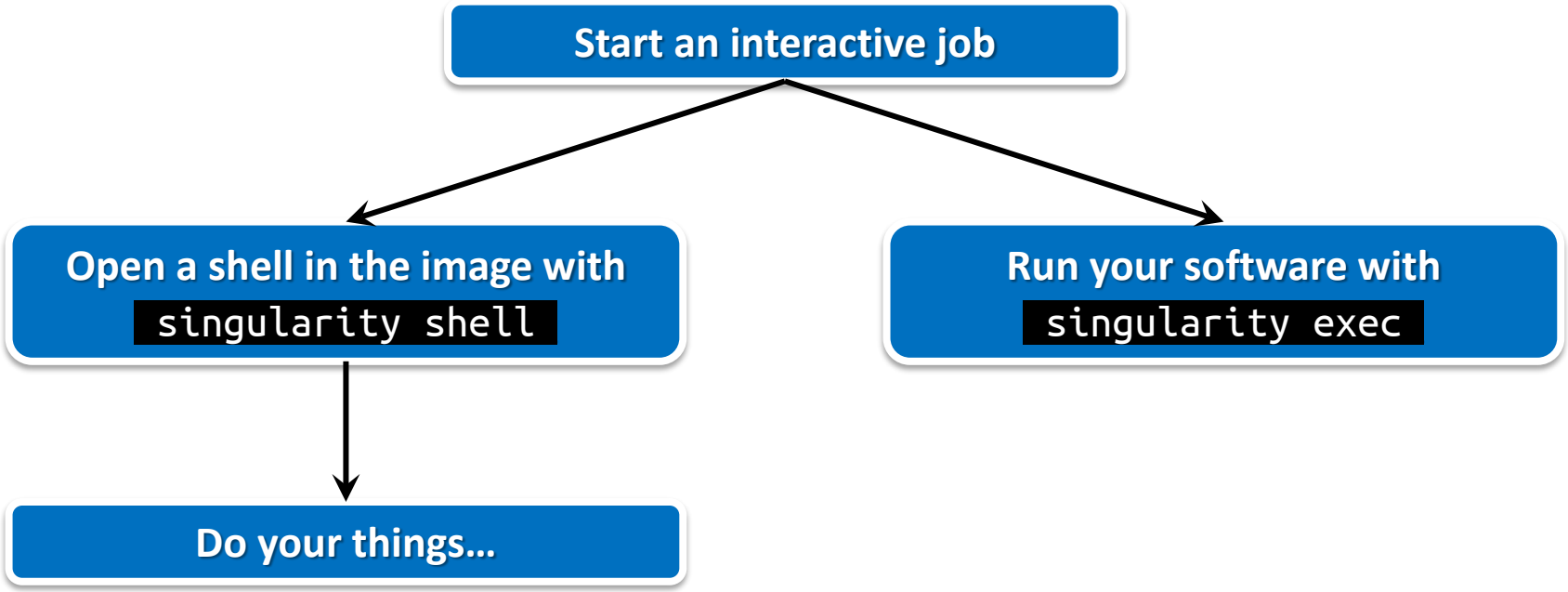
3) Run jobs with Singularity

- **Job types and commands**

Job Type	Commands	Purpose
Interactive	• `singularity shell [options] <container>` • `singularity exec [options] <container> <command>`	• Debugging & testing
Batch	• `singularity exec [options] <container> <command>`	• Production

3) Run jobs with Singularity

a) Interactive job



3) Run jobs with Singularity

b) Batch job

PBS

```
#!/bin/bash
#PBS -A <Allocation name>
#PBS -q workq
#PBS -l nodes=1:ppn=20
#PBS -l walltime=24:00:00


cd /to/work/directory
IMG=/home/admin/singularity/ubuntu-training.sif

singularity exec -B /work,/project $IMG \
  python myjob.py
```

Example

Slurm

```
#!/bin/bash
#SBATCH -A <Allocation name>
#SBATCH -p workq
#SBATCH -N 1
#SBATCH -n 64
#SBATCH -t 24:00:00

cd /to/work/directory
IMG=/home/admin/singularity/ubuntu-training.sif

singularity exec -B /work,/project $IMG \
  python myjob.py
```

Example

Summary

Summary

Syntax	Description
`singularity shell [options] <container>`	Run a prewritten script
`singularity exec [options] <container> <command>`	Execute a command in the image
`singularity run [options] <container>`	Run a prewritten script

Outlines

1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

3. Get More Container Images

- **Available images**

  – On all clusters: `/home/admin/singularity`

```
(base) [jasonli3@mike1 ~]$ ls /home/admin/singularity
abinit.v9.8.3.sif                   jax-0.4.13-gpu-jupyter.sif
alphafold-catgumag-2.2.sif          MuST-1.8.7.sif
bowtie2-2.5.1.sif                   openpose.sif
busco-5.4.7.sif                     pytorch-2.0.1-gpu-jupyter.sif
clara.400-1.sif                     rstudio-2023.03.1-446-rocky8.sif
deepmd-kit_2.0.3_cuda11.3_gpu.sif   salmon-1.10.2.sif
deepmd-kit_2.2.1_cuda11.6_gpu.sif   tensorflow-2.13.0-gpu-jupyter.sif
delft3dfm_r142632_05032023.sif      trinity-2.15.1.sif
delft3dfm_r142632.sif               ubuntu-training.sif
delft3d_r142586.sif
```

Outlines

1. Why Container?
   1) Problems
   2) Container & Singularity
2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity
3. **Get More Container Images**
   1) What you need
   2) Where to get
   3) How to get
4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

1) What you need

```
(base) [jasonli3@mike4 ~]$ ll /home/admin/singularity
total 78269032
-rwxr-xr-x 1 root singularity  4555411456 Jun  7 20:27 abinit.v9.8.3.sif
-rwxr-xr-x 1 root singularity  3167338496 Oct  3  2022 alphafold-catgumag-2.2.sif
-rwxr-xr-x 1 root singularity   118206464 Sep 11 11:00 bowtie2-2.5.1.sif
-rwxr-xr-x 1 root singularity   822177792 Aug 22 09:00 busco-5.4.7.sif
-rwxr-xr-x 1 root singularity  2155438080 Nov  9  2022 clara.400-1.sif
-rwxr-xr-x 1 root singularity  3285417984 Jun  6 13:13 deepmd-kit_2.0.3_cuda11.3_gpu.sif
-rwxr-xr-x 1 root singularity  3390902272 Jun  5 20:34 deepmd-kit_2.2.1_cuda11.6_gpu.sif
-rwxr-xr-x 1 root singularity  9305526272 May  3 12:56 delft3dfm_r142632_05032023.sif
-rwxr-xr-x 1 root singularity 11812220928 Jun 27 23:50 delft3dfm_r142632.sif
-rwxr-xr-x 1 root singularity 11594326016 Jul 26 00:05 delft3d_r142586.sif
-rwxr-xr-x 1 root singularity  5152387072 Sep 15 23:52 jax-0.4.13-gpu-jupyter.sif
-rwxr-xr-x 1 root singularity  7603736576 Jul 20 14:57 MuST-1.8.7.sif
-rwxr-xr-x 1 root singularity  5322223616 Jun 20 14:26 openpose.sif
-rwxr-xr-x 1 root singularity  4016316416 Aug  7 23:56 pytorch-2.0.1-gpu-jupyter.sif
-rwxr-xr-x 1 root singularity   911499264 May 25 09:48 rstudio-2023.03.1-446-rocky8.sif
-rwxr-xr-x 1 root singularity    42639360 Sep 21 12:22 salmon-1.10.2.sif
-rwxr-xr-x 1 root singularity  4079706112 Aug  8 00:32 tensorflow-2.13.0-gpu-jupyter.sif
-rwxr-xr-x 1 root singularity  2739630080 Aug 30 19:40 trinity-2.15.1.sif
-rwxr-xr-x 1 root singularity    71102464 Sep  7 11:50 ubuntu-training.sif
```

1) What you need

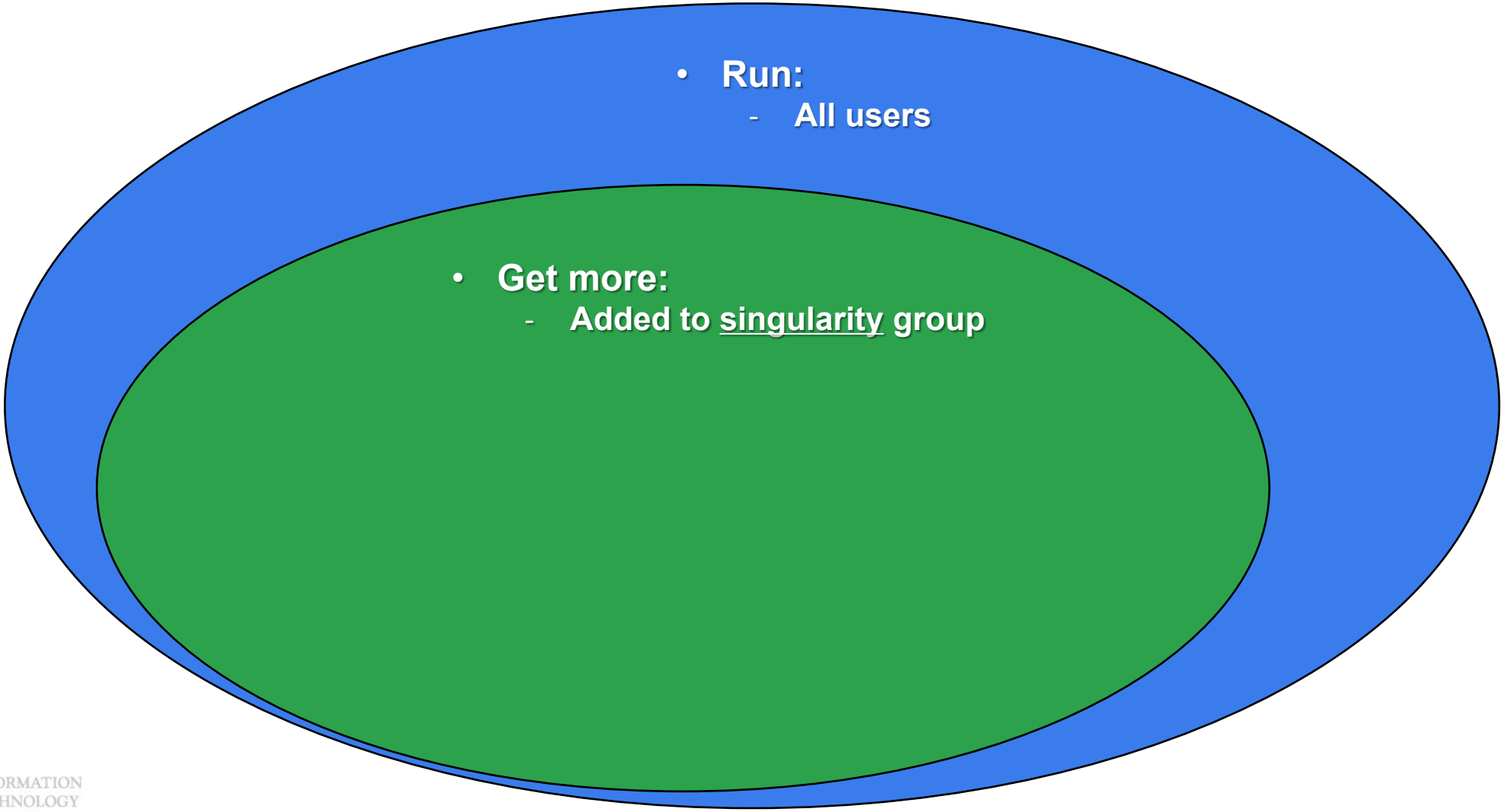
```
(base) [jasonli3@mike4 ~]$ ll /home/admin/singularity
total 78269032
-rwxr-xr-x 1 root singularity  4555411456 Jun  7 20:27 abinit.v9.8.3.sif
-rwxr-xr-x 1 root singularity  3167338496 Oct  3  2022 alphafold-catgumag-2.2.sif
-rwxr-xr-x 1 root singularity   118206464 Sep 11 11:00 bowtie2-2.5.1.sif
-rwxr-xr-x 1 root singularity   822177792 Aug 22 09:00 busco-5.4.7.sif
-rwxr-xr-x 1 root singularity  2155438080 Nov  9  2022 clara.400-1.sif
-rwxr-xr-x 1 root singularity  3285417984
-rwxr-xr-x 1 root singularity  3390902271
-rwxr-xr-x 1 root singularity  9305526271
-rwxr-xr-x 1 root singularity 11812220928
-rwxr-xr-x 1 root singularity 11594326016
-rwxr-xr-x 1 root singularity  5152387071
-rwxr-xr-x 1 root singularity  7603736576
-rwxr-xr-x 1 root singularity  5322223616
-rwxr-xr-x 1 root singularity  4016316416 Aug  7 23:56 pytorch-2.0.1-gpu-jupyter.sif
-rwxr-xr-x 1 root singularity   911499264 May 25 09:48 rstudio-2023.03.1-446-rocky8.sif
-rwxr-xr-x 1 root singularity    42639360 Sep 21 12:22 salmon-1.10.2.sif
-rwxr-xr-x 1 root singularity  4079706112 Aug  8 00:32 tensorflow-2.13.0-gpu-jupyter.sif
-rwxr-xr-x 1 root singularity  2739630080 Aug 30 19:40 trinity-2.15.1.sif
-rwxr-xr-x 1 root singularity    71102464 Sep  7 11:50 ubuntu-training.sif
```

Singularity images must belong to "**singularity**" group to run on our clusters!

1) What you need

To …	What you need …
Run an Existing Container Image	• Access to our HPC systems - An active account - An active allocation
Get More Container Images	• (Everything above) • Added to **Singularity** group - Needed to change **group ownership** - Send email to sys-help@loni.org to request

1) What you need



Outlines

1. Why Container?
   1) Problems
   2) Container & Singularity
2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity
3. **Get More Container Images**
   1) What you need
   2) Where to get
   3) How to get
4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

2) Where to get

- **You can get container images from a lot of places**
  - ➢ **Not that you should!**
- **Concerns?**
  - **Reliability** (some third-party or untested images may not work)
  - **Security risk** (some untrustworthy publishers may pack something you don't know about)
- **Solution**
  - Always get from a source that **you can trust**

2) Where to get

- **Tier 1: Developer release (official release)**
  - On software's official website, look for "**Docker**" / "**Singularity**" / "**Container**" / etc.
  - E.g., Tensorflow, Trinity, Salmon
- **Tier 2: Trustworthy third party**

Name	Notes
Biocontainers	• https://biocontainers-edu.readthedocs.io/en/latest/ • For biology
Nvidia NGC	• https://catalog.ngc.nvidia.com/containers • For Nvidia GPU
Bitnami	• https://bitnami.com/stacks/containers • By VmWare
Docker Hub **Quay.io**	• https://hub.docker.com/ & https://quay.io/ • Don't just trust everything you see there! • Look for Docker Official Image or Verified Publisher • Avoid third-party publishers that you don't know

Outlines

1. Why Container?
   1) Problems
   2) Container & Singularity
2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity
3. **Get More Container Images**
   1) What you need
   2) Where to get
   3) How to get
4. Build Your Own Container Image
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

3) How to get

- **Steps:**

  a) Step 1: Pull the image

  b) Step 2: Change group ownership

3) How to get

a) Step 1: Pull the image

Syntax		Description
`singularity pull <source>`		Pull an image from source

`<source>`	`docker://container[:tag]`
	• (Compare to Docker command)
	`docker pull container[:tag]`
		Pull a Docker container and convert to Singularity
		• Many software's official container release is in Docker form (may or may not on Docker Hub)

	`http://www.myexample.com/container_image.sif`	Download an image file from a web source

3) How to get

a) Step 1: Pull the image

Syntax		Description
`singularity build <target> <source>`		Build an image from source (Advanced)
`<source>`	`docker://container[:tag]`	Build from a Docker container
	`container_image.sif`	Build from a local image file
	`container_sandbox/`	Build from a local **sandbox** (A directory form of a container)
	`container_recipe.def`	Build from a **recipe** (an instruction script of how to build an image)

3) How to get

a) Step 1: Pull the image

Syntax	Description
`singularity pull [options] [target] <source>`	Simple pull
`singularity build [options] <target> <source>`	Advanced build command

3) How to get

b) Step 2: Change group ownership

– What if you do not?

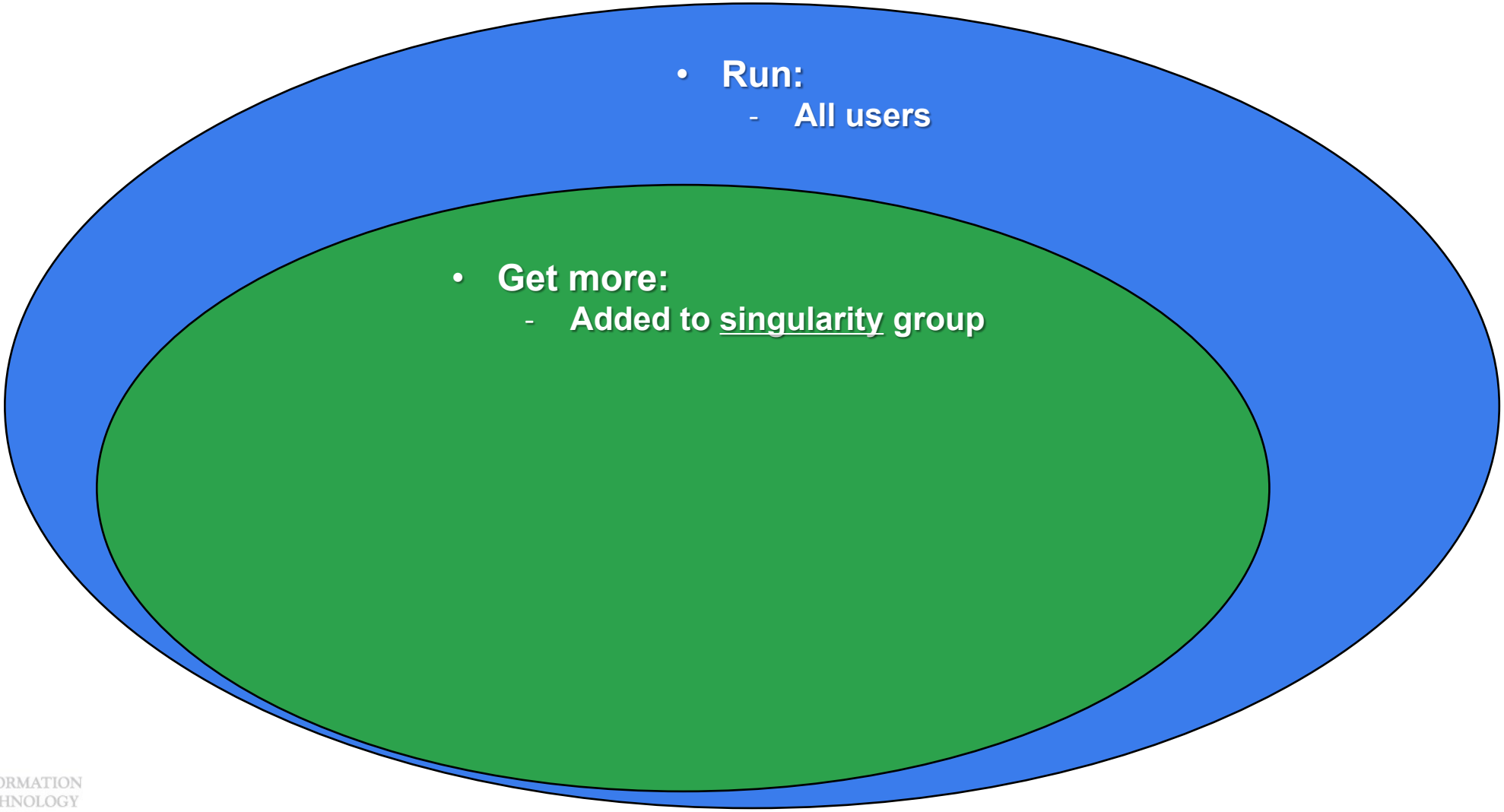
```
FATAL:   singularity image is not owned by required group(s)
```

– To solve it, run this:

```
$ chgrp singularity <container>
```

* You must be added to singularity group to finish this step

Summary



Summary

- **Steps:**

a) Step 1: Pull the image

Syntax	Description
`singularity pull [options] [target] <source>`	Simple pull
`singularity build [options] <target> <source>`	Advanced build command

b) Step 2: Change group ownership

Outlines

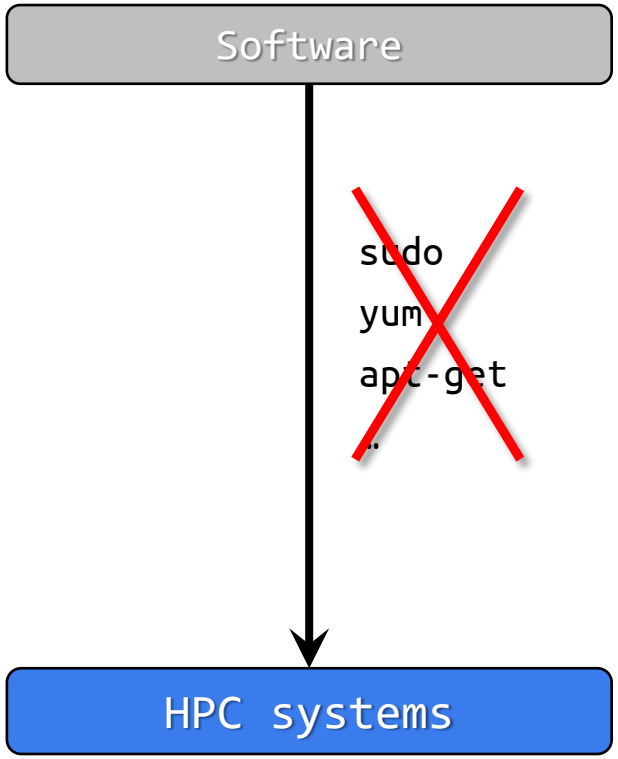
1. Why Container?
   1) Problems
   2) Container & Singularity
2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity
3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get
4. **Build Your Own Container Image**
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

4. Build Your Own Container Image

- **Scenarios:**
  - I did not find any container release. Need to DIY.
  - Installation can be easily done using `sudo apt` or `sudo yum` if I have the permission.
  - I found a container, but need to make changes (e.g., adding something else).

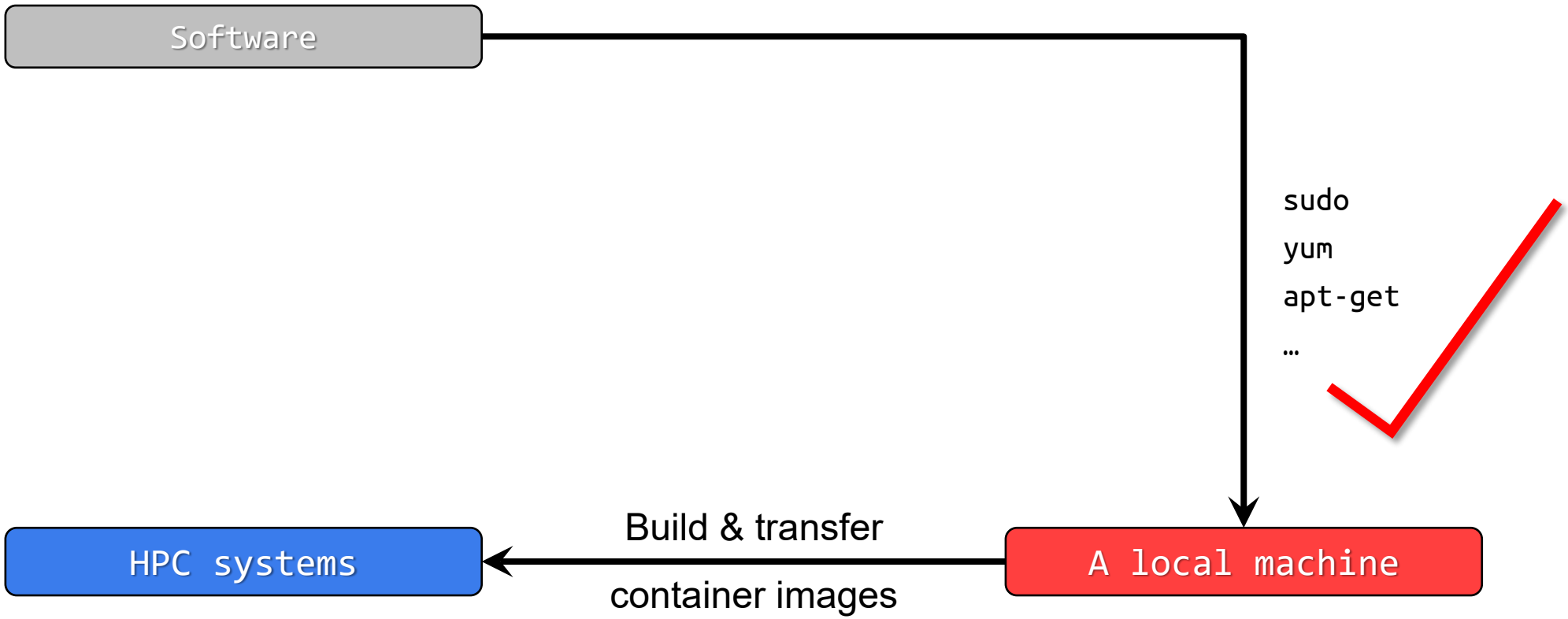
4. Build Your Own Container Image

- **Idea**



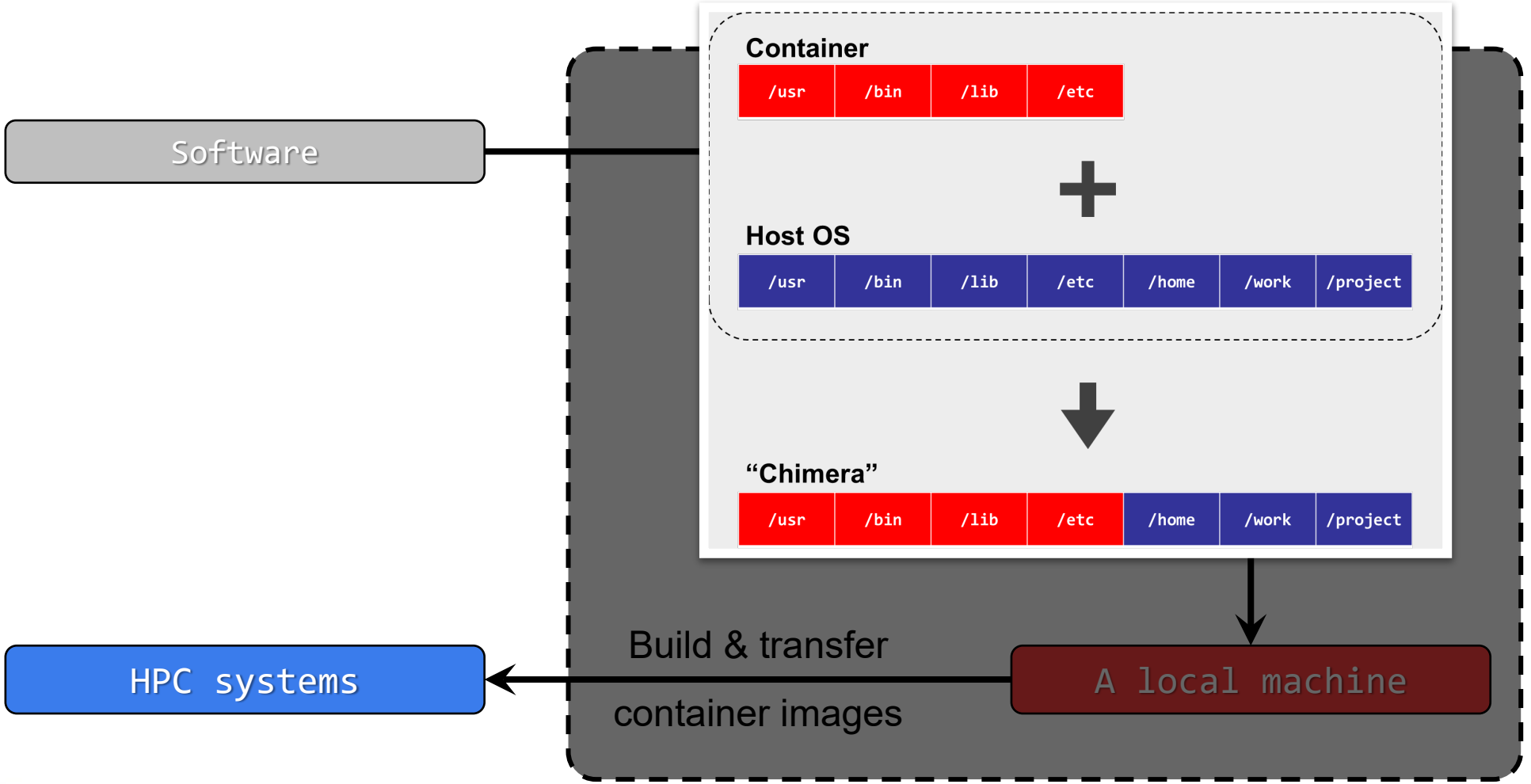
4. Build Your Own Container Image

- **Idea**



4. Build Your Own Container Image

- **Idea**



Outlines

1. Why Container?
   1) Problems
   2) Container & Singularity
2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity
3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get
4. **Build Your Own Container Image**
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

1) What you need

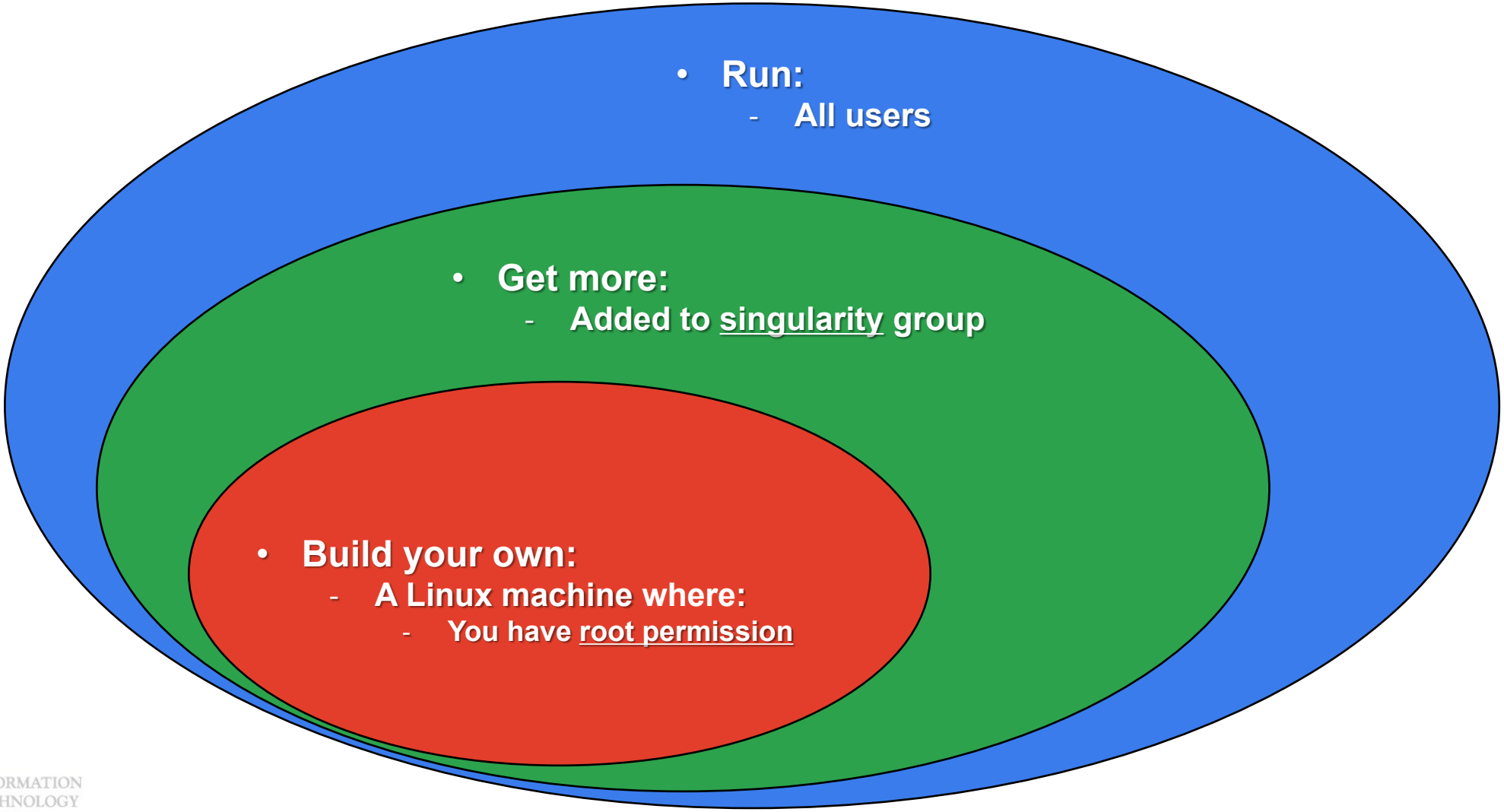
To …	What you need …

Run an Existing Container Image	• Access to our HPC systems
	  - An active account
	  - An active allocation

Get More Container Images	• (Everything above)
	• Added to **Singularity** group
	  - Needed to change **group ownership**
	  - Send email to sys-help@loni.org to request

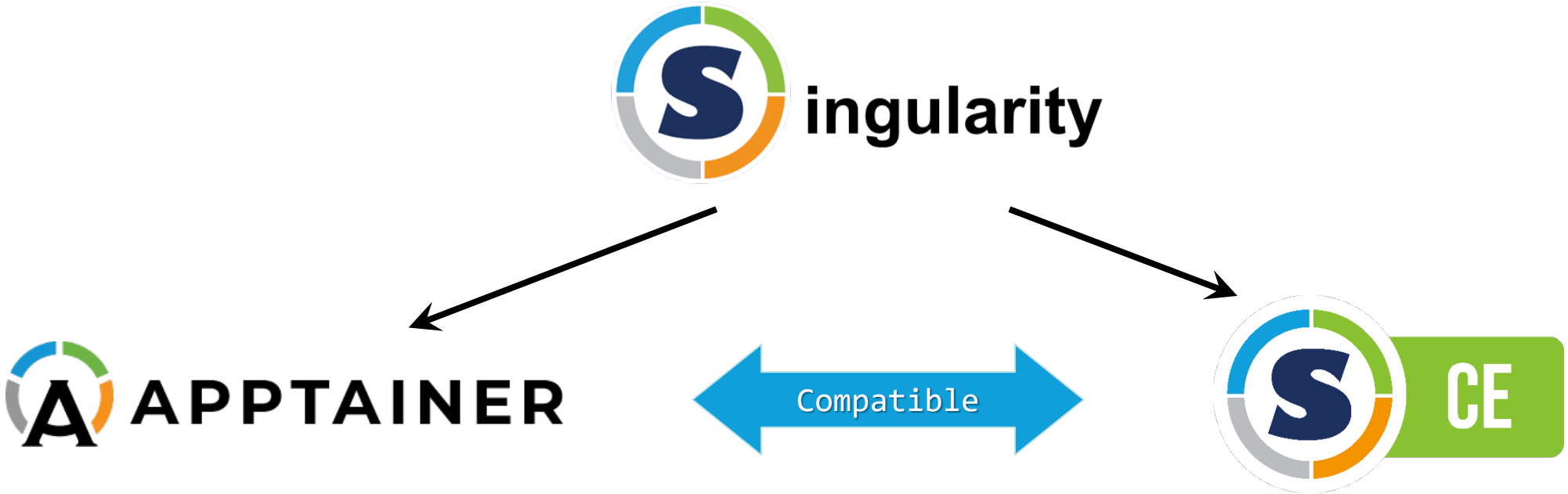
Build Your Own Container Image	• (Everything above) • A local Linux machine, where: - You have **root permission**

1) What you need



1) What you need

- **Install Singularity**



- Joined Linux Foundation
- Easier installation

- Community supported
- Installed on our clusters

[1] https://apptainer.org/docs/admin/main/installation.html
[2] https://docs.sylabs.io/guides/3.8/admin-guide/installation.html

Outlines

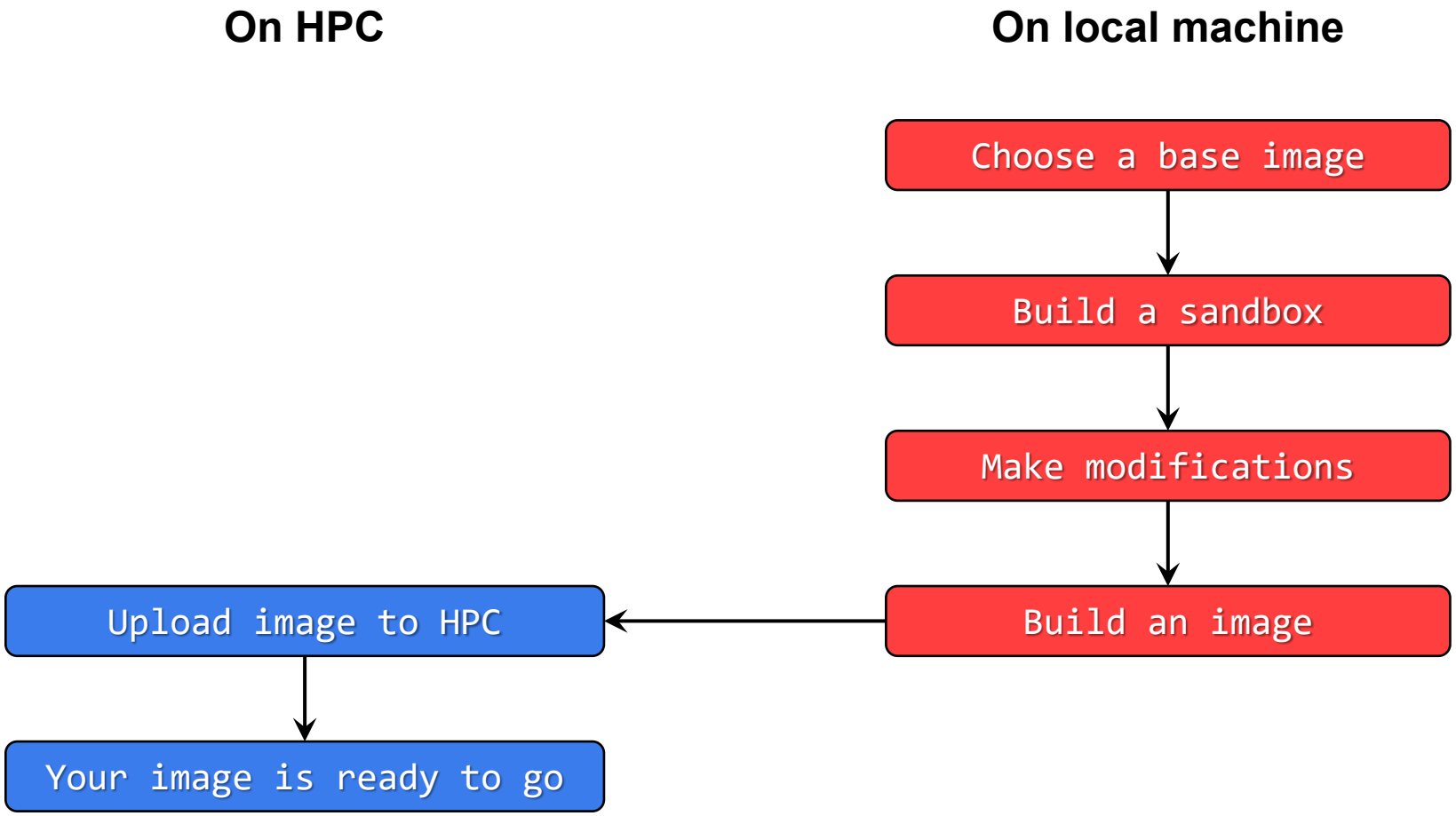
1. Why Container?
   1) Problems
   2) Container & Singularity

2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get

4. **Build Your Own Container Image**
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

2) Typical workflow



2) Typical workflow

a) Choose a base image

Common choices	Typical scenarios
A minimum, "mint" OS (e.g., Ubuntu, Rocky, Debian, …)	• You cannot find an existing image with the software you need, and need to install from the scratch. • You need to build a minimum size image
A container with software already installed (e.g., TensorFlow, PyTorch, …)	• You need to modify an existing working image (e.g., add a Python module to Tensorflow image)

b) Build a sandbox

– What's a **sandbox** ?

- A **directory** form (unlike a single image file) of a container
- Allows modification

2) Typical workflow

b) Build a sandbox

```
$ singularity build [options] <target> <source>
```

<source>	docker://container[:tag]	Build from a Docker container
	container_image.sif	Build from a local image file
	container_sandbox/	Build from a local **sandbox** (A directory form of a container)
	container_recipe.def	Build from a **recipe** (an instruction script of how to build an image)

2) Typical workflow

b) Build a sandbox

```
$ singularity build --sandbox [options] <target> <source>
```

<source>	docker://container[:tag]	Build from a Docker container
	container_image.sif	Build from a local image file
	container_sandbox/	Build from a local **sandbox** (A directory form of a container)
	container_recipe.def	Build from a **recipe** (an instruction script of how to build an image)

2) Typical workflow

c) Make modifications

```
$ singularity shell [options] <container>
```

2) Typical workflow

c) Make modifications

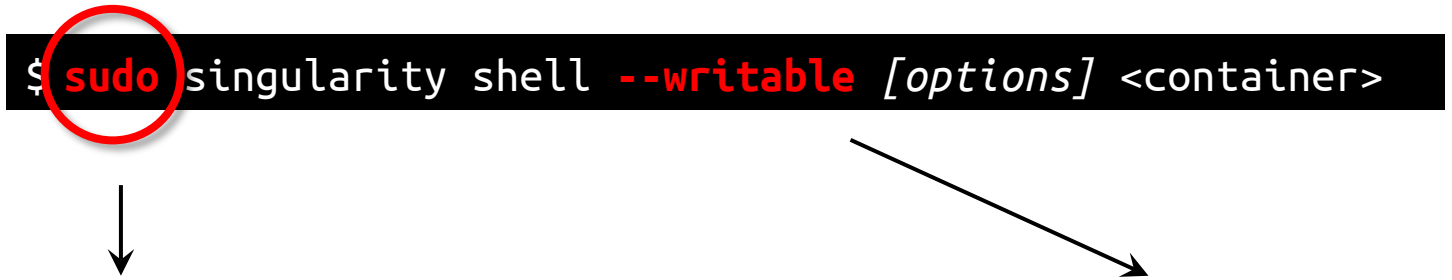
```
$ singularity shell --writable [options] <container>
```

i. Allows **writing** to the sandbox

   – Without it, just like running a regular container image

2) Typical workflow

c) Make modifications



ii. Run the container as **root**

- Grants root privilege in container
- Needed in most cases
- Technically not required, but cannot run things like `sudo apt` or `sudo yum` without it

i. Allows **writing** to the sandbox

- Without it, just like running a regular container image

2) Typical workflow

c) Make modifications

```
$ sudo singularity shell --writable [options] <container>
Singularity>
Singularity> apt update
Singularity> apt install ...
```

2) Typical workflow

d) Build an image from sandbox

```
$ singularity build [options] <target> <source>
```

<source>	docker://container[:tag]	Build from a Docker container
	container_image.sif	Build from a local image file
	container_sandbox/	Build from a local **sandbox** (A directory form of a container)
	container_recipe.def	Build from a **recipe** (an instruction script of how to build an image)

2) Typical workflow

d) Build an image from sandbox

```
$ sudo singularity build [options] <target> <source>
```

Modify with “**sudo**”? Build with “**sudo**”!

2) Typical workflow

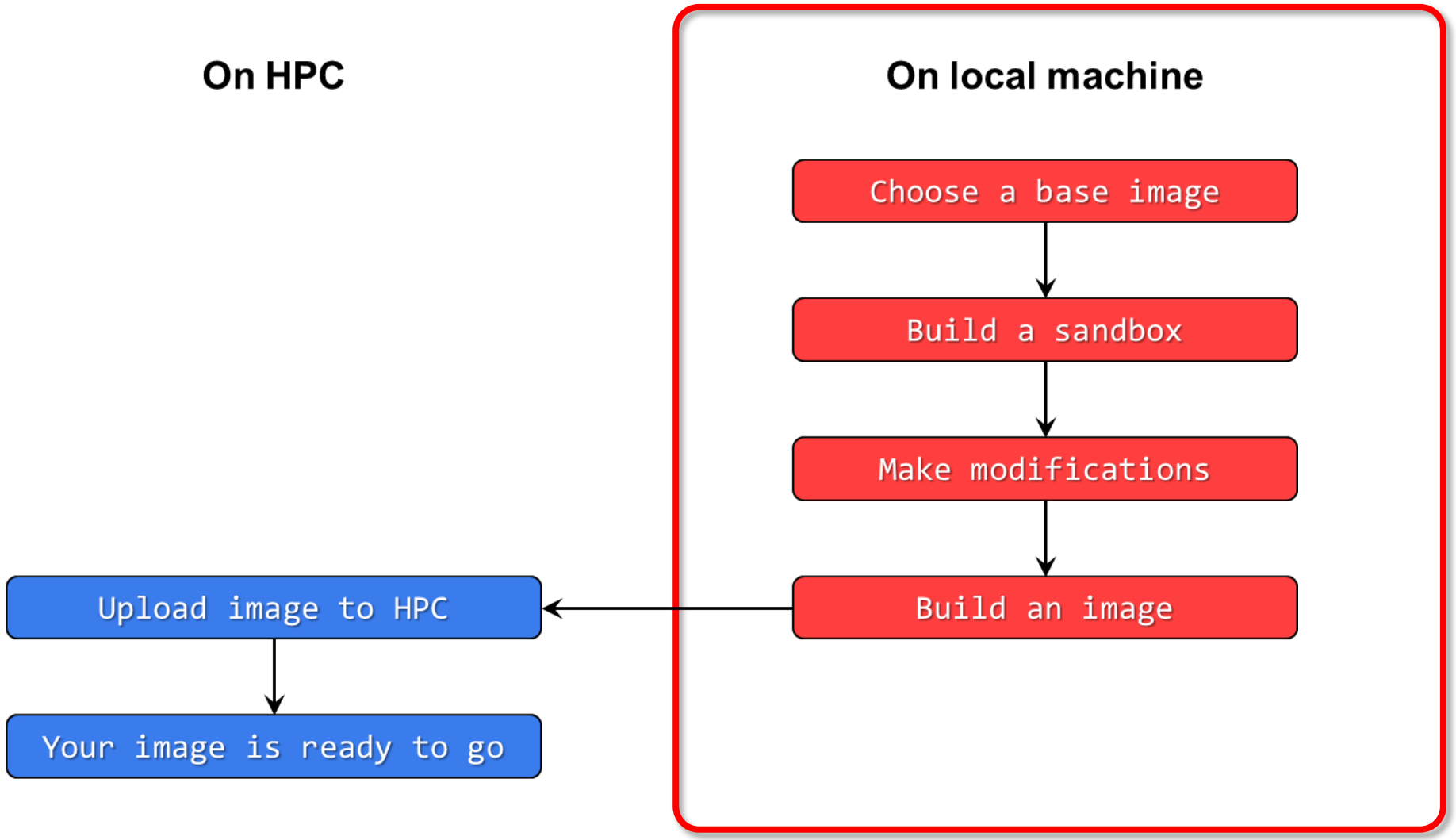
- **Quick recap**

To …	You need to …
Build a sandbox	`$ singularity build --sandbox ...`
Modify a sandbox	`$ sudo singularity shell --writable ...`
Build an image from sandbox	`$ sudo singularity build ...`

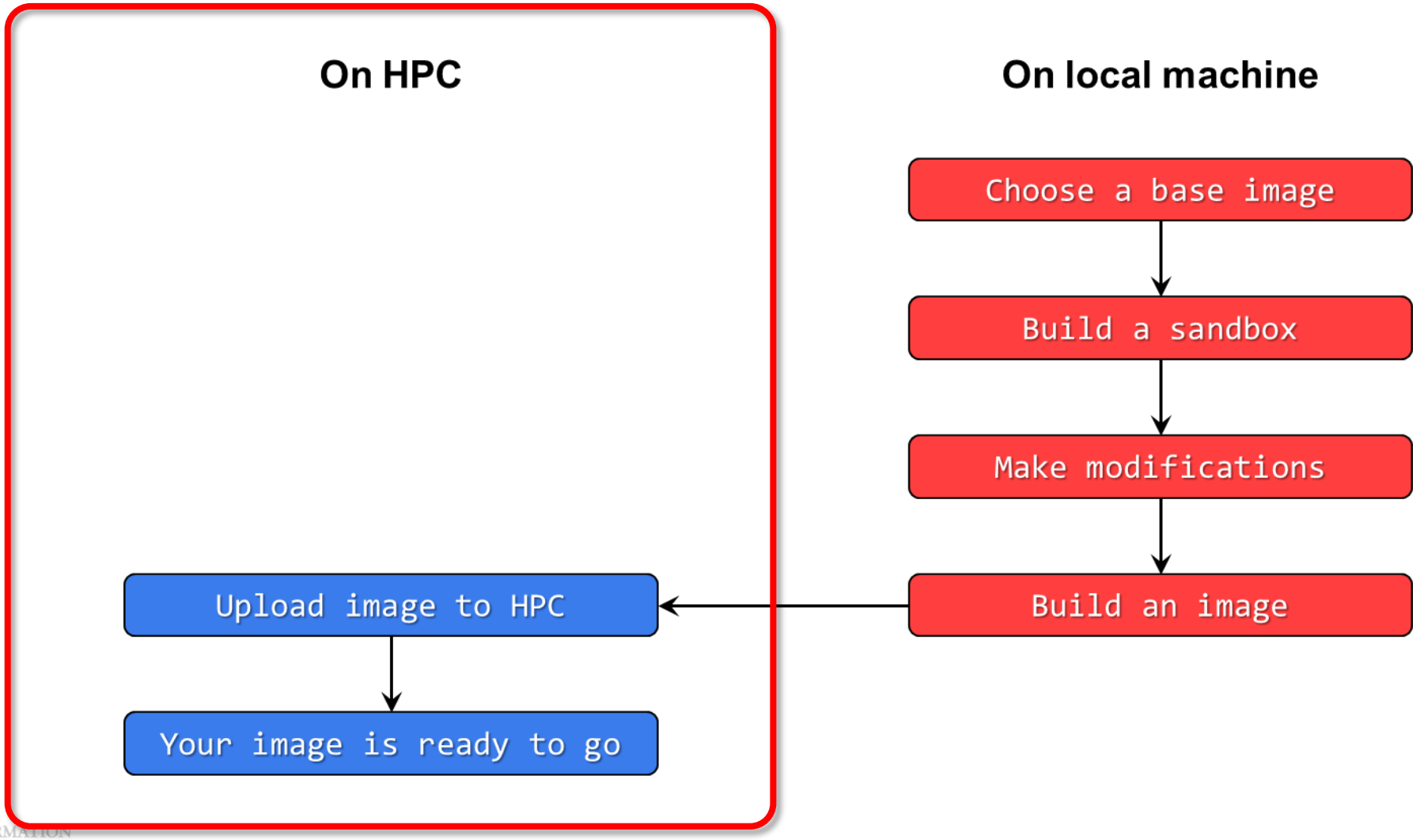
e) Upload image to HPC and run

Now! The moment of truth!

2) Typical workflow



2) Typical workflow

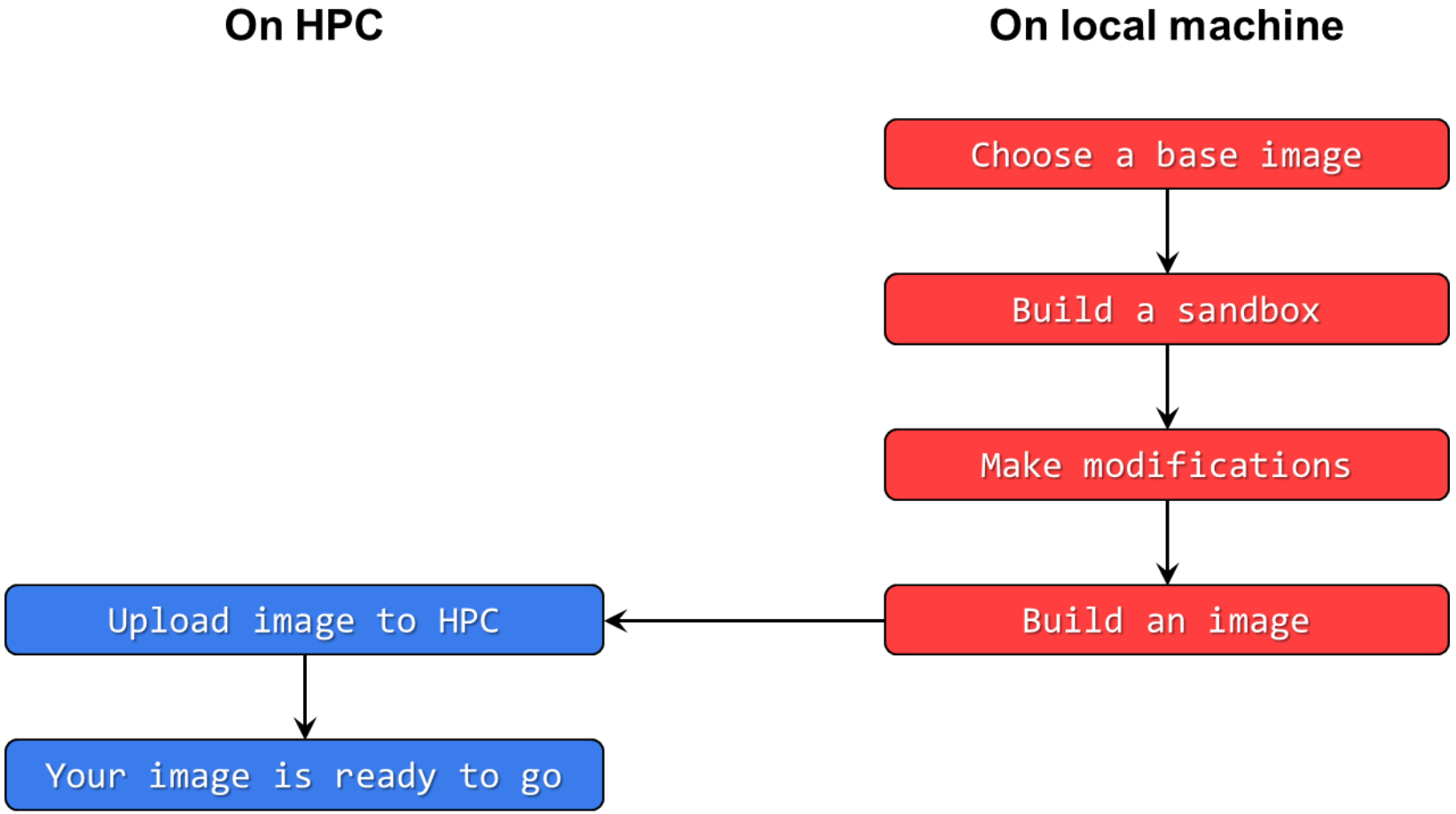


Outlines

1. Why Container?
   1) Problems
   2) Container & Singularity
2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity
3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get
4. **Build Your Own Container Image**
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

3) Make it easier - Recipe

- **Why?**



3) Make it easier - Recipe

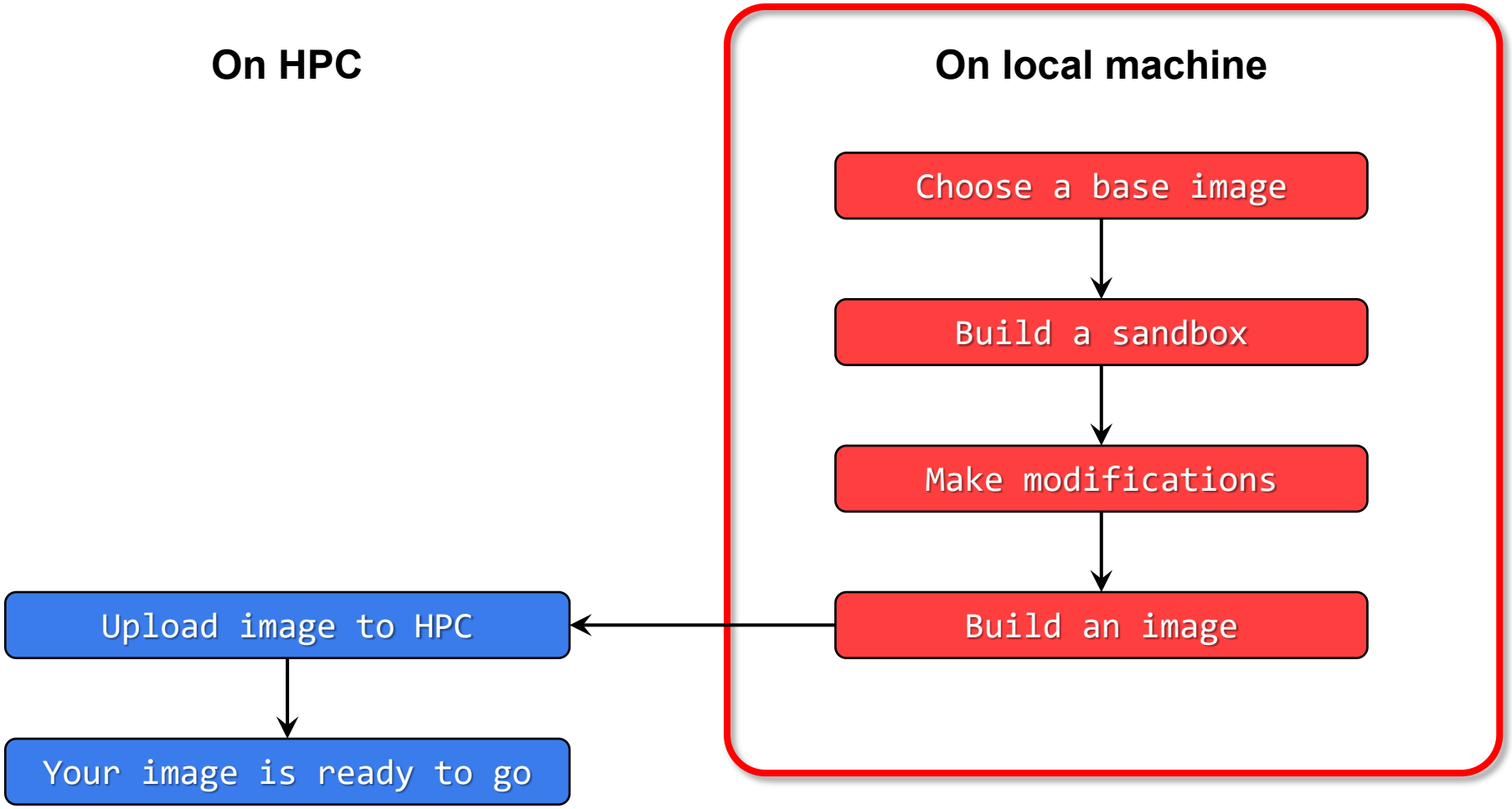
- **Why?**

Pros	Cons
• Flexibility	• Repeatability • Minimizing image size

- **Solution:**
  - **Recipe**: A text file containing instructions to build a container

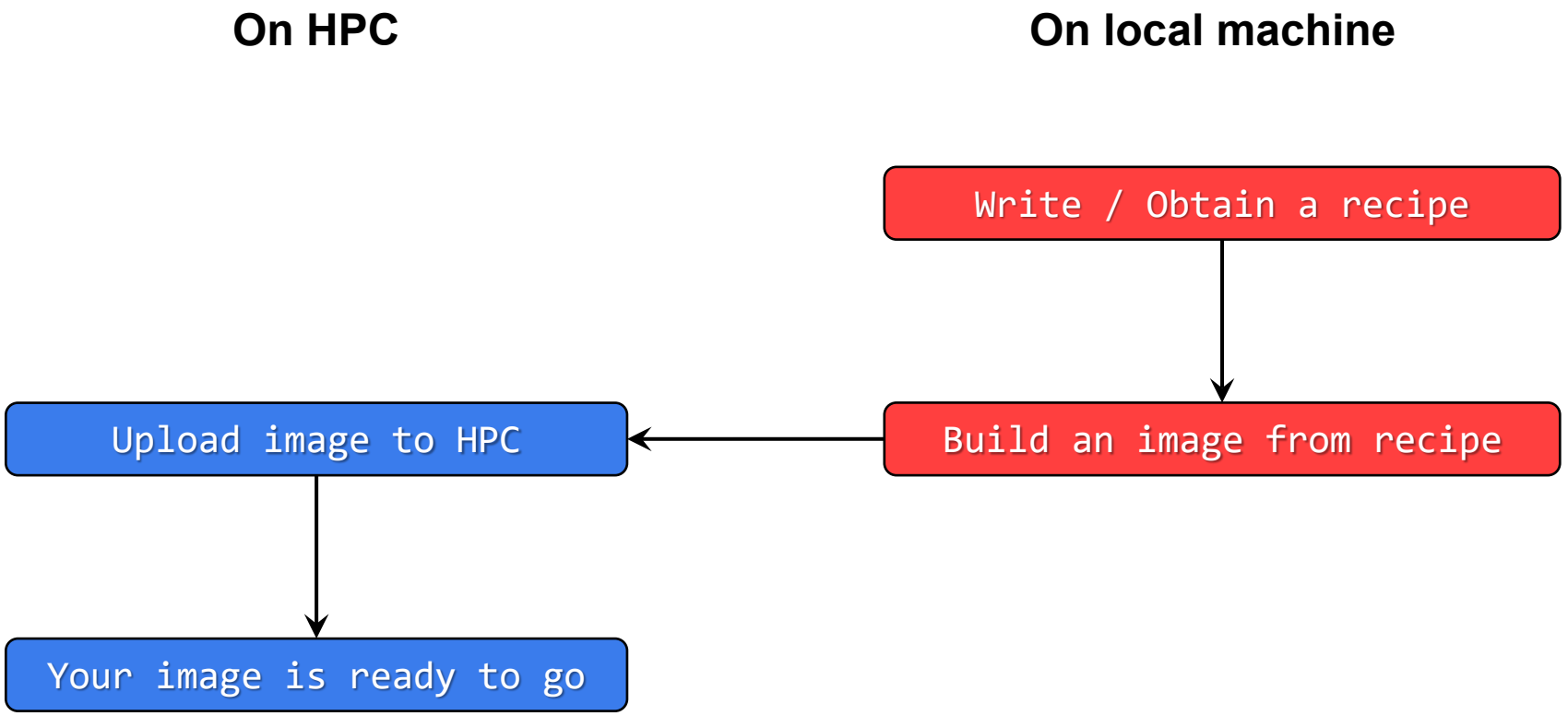
3) Make it easier - Recipe

- **Why?**



3) Make it easier - Recipe

- **Why?**



3) Make it easier - Recipe

a) What does a recipe look like?

ruby.def

```
BootStrap: docker
From: ubuntu:latest

%labels
Author       Jason Li
Description  A container with Ruby installed

%post
apt update
apt install -y ruby

%environment
export MYENV="Some environmental variable"

%runscript
ruby -e "puts 'Hello from container!'"
```

3) Make it easier - Recipe

a) What does a recipe look like?

ruby.def

```
BootStrap: docker
From: ubuntu:latest

%labels
Author        Jason Li
Description   A container with Ruby installed

%post
apt update
apt install -y ruby

%environment
export MYENV="Some environmental variable"

%runscript
ruby -e "puts 'Hello from container!'"
```

Header
- Base image info (how, where, what to pull)

3) Make it easier - Recipe

a) What does a recipe look like?

ruby.def

```
BootStrap: docker
From: ubuntu:latest

%labels
Author        Jason Li
Description   A container with Ruby installed

%post
apt update
apt install -y ruby

%environment
export MYENV="Some environmental variable"

%runscript
ruby -e "puts 'Hello from container!'"
```

Label

- Container information (write whatever you want)

3) Make it easier - Recipe

a) What does a recipe look like?

ruby.def

```
BootStrap: docker
From: ubuntu:latest

%labels
Author        Jason Li
Description   A container with Ruby installed

%post
apt update
apt install -y ruby

%environment
export MYENV="Some environmental variable"

%runscript
ruby -e "puts 'Hello from container!'"
```

Post
- Commands to execute after the base image is pulled

3) Make it easier - Recipe

a) What does a recipe look like?

ruby.def

```
BootStrap: docker
From: ubuntu:latest

%labels
Author        Jason Li
Description   A container with Ruby installed

%post
apt update
apt install -y ruby

%environment
export MYENV="Some environmental variable"

%runscript
ruby -e "puts 'Hello from container!'"
```

Environment

- Define environmental variables every time the container is executed

3) Make it easier - Recipe

a) What does a recipe look like?

ruby.def

```
BootStrap: docker
From: ubuntu:latest

%labels
Author        Jason Li
Description   A container with Ruby installed

%post
apt update
apt install -y ruby

%environment
export MYENV="Some environmental variable"

%runscript
ruby -e "puts 'Hello from container!'"
```

Runscript
- Commands to be run with `singularity run`

3) Make it easier - Recipe

a) What does a recipe look like?

ruby.def

```
BootStrap: docker
From: ubuntu:latest

%labels
Author        Jason Li
Description   A container with Ruby installed

%post
apt update
apt install -y ruby

%environment
export MYENV="Some environmental variable"

%runscript
ruby -e "puts 'Hello from container!'"
```

3) Make it easier - Recipe

a) What does a recipe look like?

ruby.def

```
BootStrap: docker
From: ubuntu:latest

%labels
Author        Jason Li
Description   A container with Ruby installed

%post
apt update
apt install -y ruby

%environment
export MYENV="Some environmental variable"

%runscript
ruby -e "puts 'Hello from container!'"
```

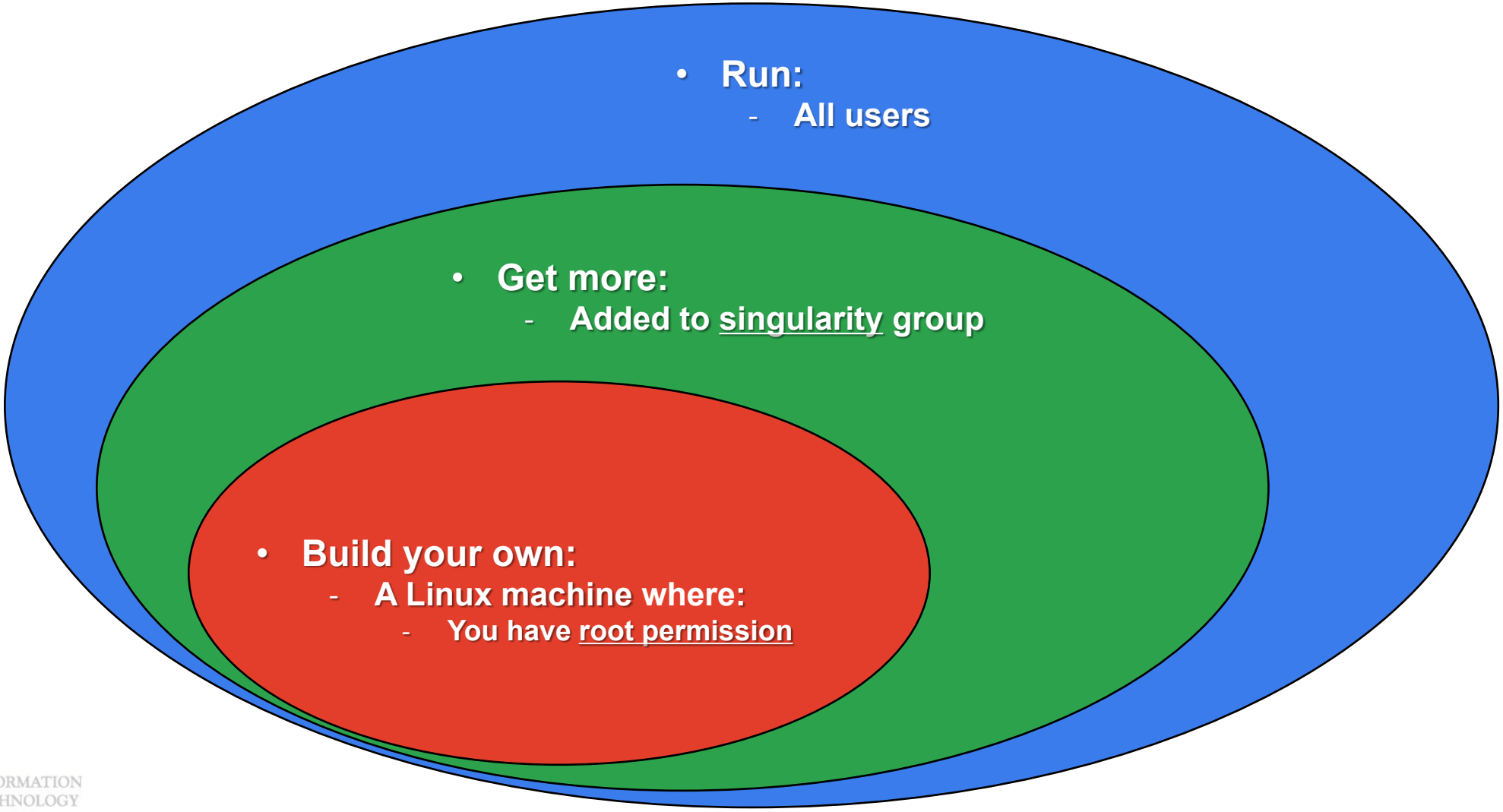
3) Make it easier - Recipe

b) Build the recipe

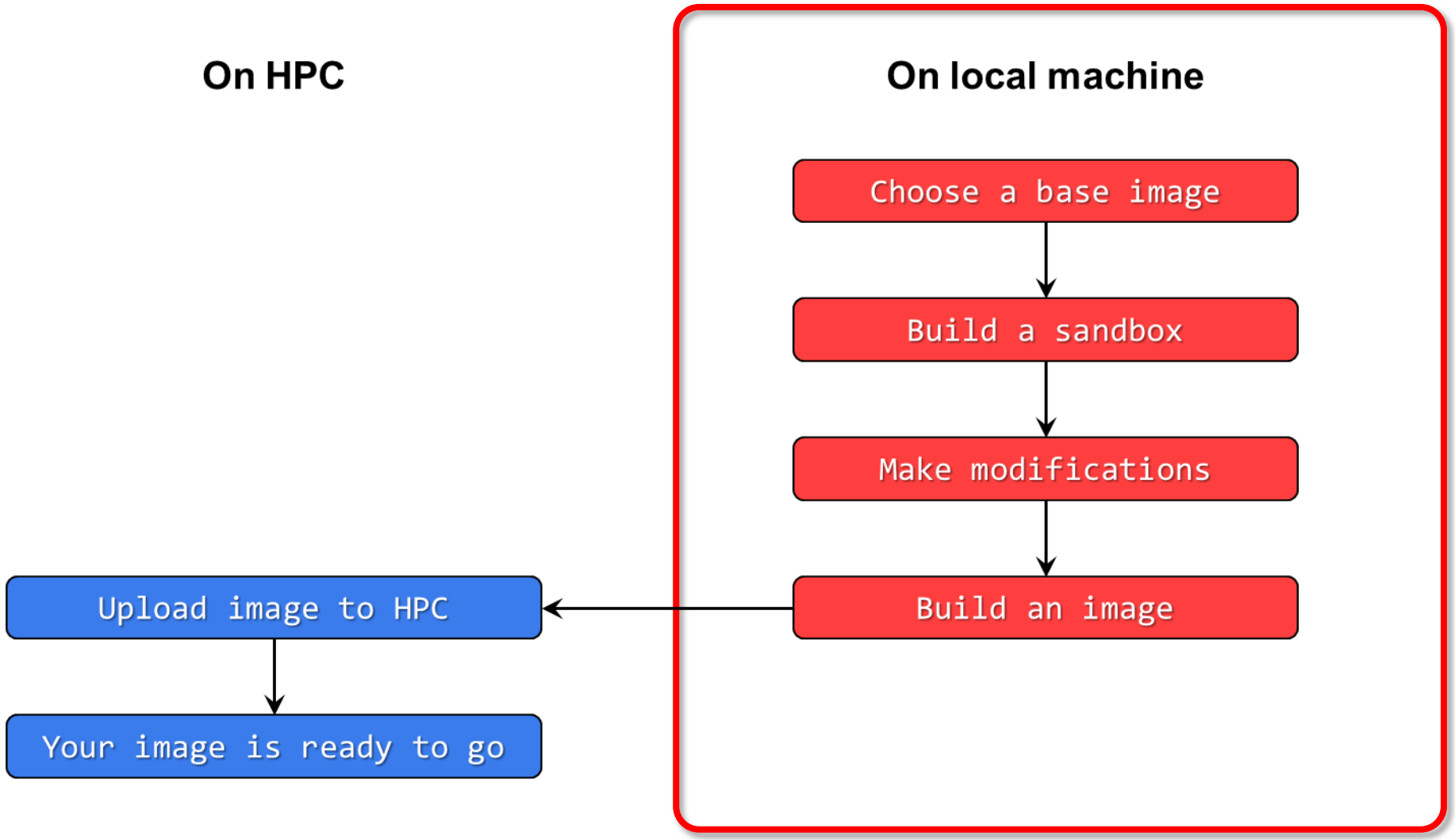
```
$ singularity build [options] <target> <source>
```

<source>	docker://container[:tag]	Build from a Docker container
	container_image.sif	Build from a local image file
	container_sandbox/	Build from a local **sandbox** (A directory form of a container)
	container_recipe.def	Build from a **recipe** (an instruction script of how to build an image)

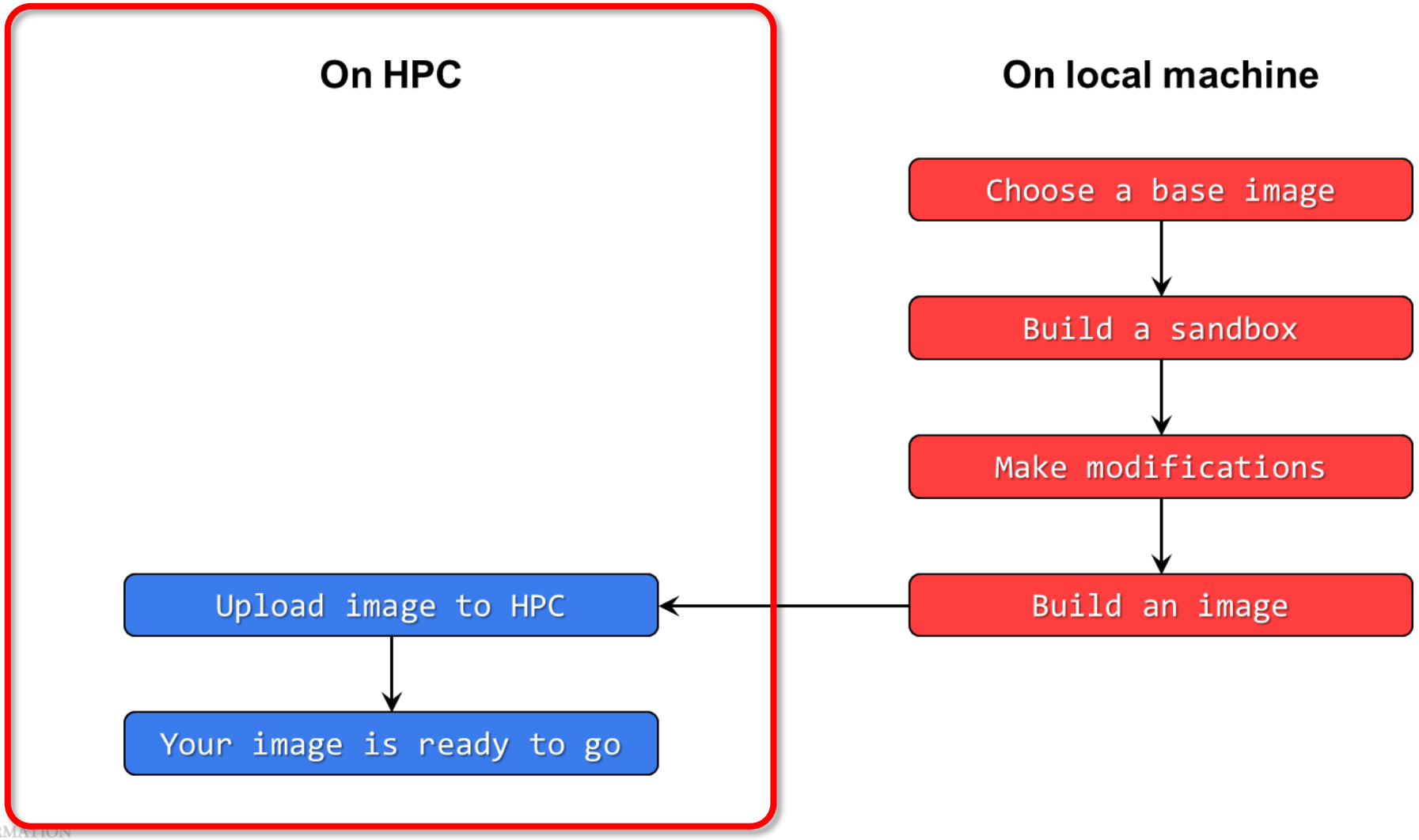
Summary


• Run:
- All users
• Get more:
- Added to singularity group
• Build your own:
- A Linux machine where:
- You have root permission

Summary


On HPC
On local machine
Choose a base image
Build a sandbox
Make modifications
Build an image
Upload image to HPC
Your image is ready to go

Summary


On HPC
On local machine
Choose a base image
Build a sandbox
Make modifications
Build an image
Upload image to HPC
Your image is ready to go

Conclusion

Conclusion

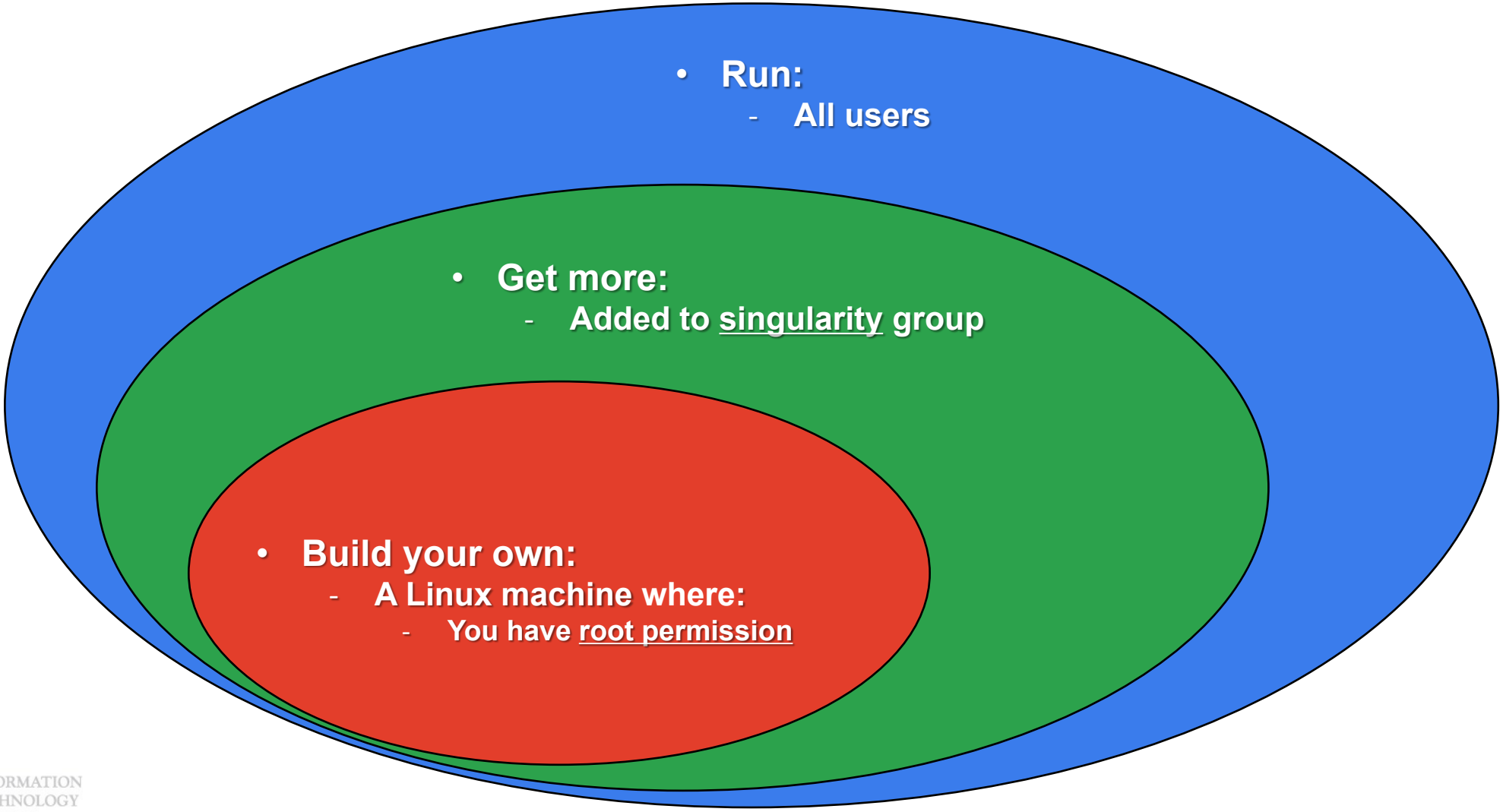
1. **Why Container?**
   1) Problems
   2) Container & Singularity

2. **Run an Existing Container Image**
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

3. **Get More Container Images**
   1) What you need
   2) Where to get
   3) How to get

4. **Build Your Own Container Image**
   1) What you need
   2) Typical workflow
   3) Make it easier - Recipe

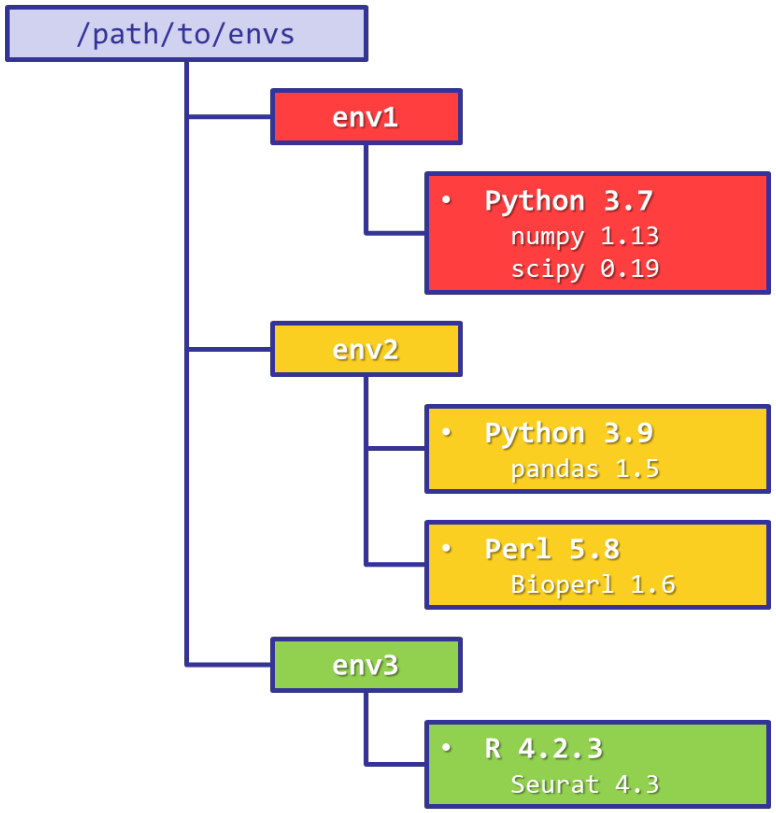
Take home message

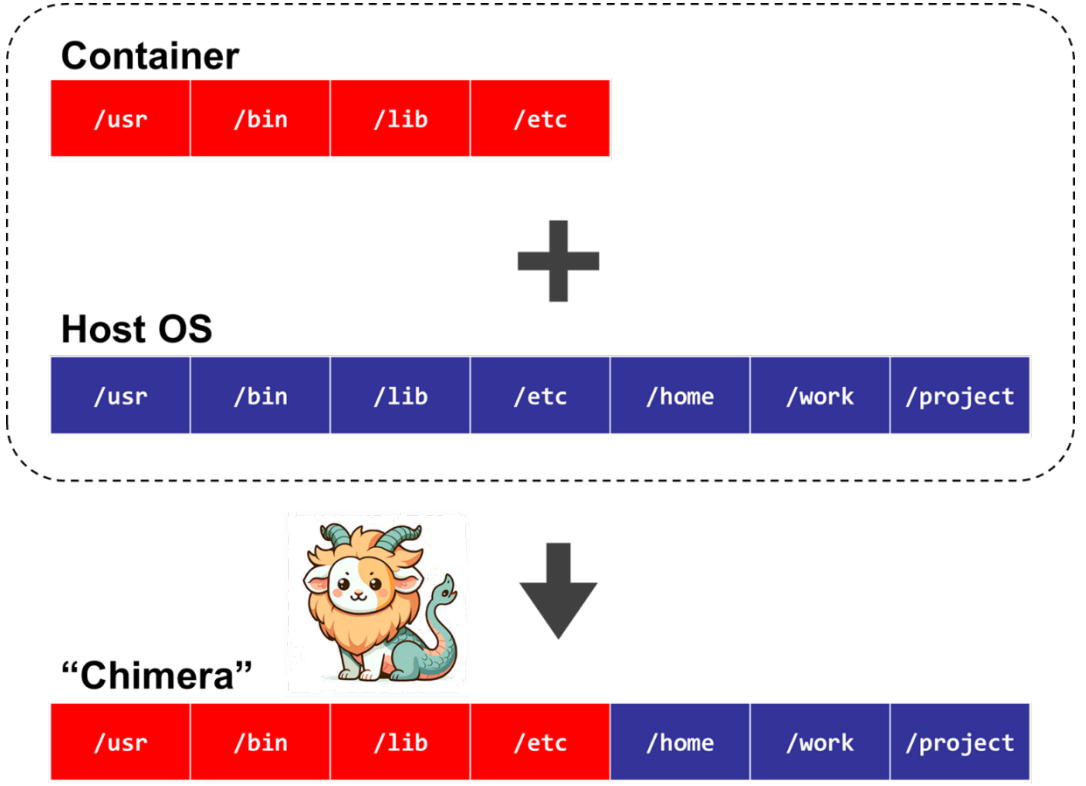


To conclude our mini series…

Conda vs Singularity

- **Virtual Environment** v.s. **Container** ?





Conda vs Singularity

	Conda / Virtual Environments	Singularity / Containers
Availability	All users	All users, but may need additional things
Self-contained	Yes	Yes
Isolated	Yes (but still accessible from outside)	Perfect (completely isolated from outside)
Editability	Yes	No (Must create a new image)
Disk usage	Large	Smaller
Portability	Possible (with .yml recipe)	Great (just copy-paste one file)
Security	Fair	Good
Ease of use	Good	May require a little more understanding

Conda vs Singularity

	Conda / Virtual Environments	Singularity / Containers
Good for	• Less hassle to create and install software from scratch • If you need to frequently make modifications	• Less hassle if the developer releases a working container • If you don't need to make changes after it is created • Portability • Reduce disk usage • Your system admins yelled at you about security risk

Contact us

- **Contact user services**
  - Email Help Ticket: sys-help@loni.org
  - Telephone Help Desk: +1 (225) 578-0900

“Commercial” time!

- **Are you tired of wring the long, tedious singularity commands?**

```
$ singularity exec --nv -B /work,/project,/usr/local/package \
    /home/admin/singularity/ubuntu-training.sif \
    python helloworld.py
```

“Commercial” time!

- Try **SIMPLE-MOD** !
  - https://github.com/lsuhpchelp/SIMPLE-MOD
  - A GUI tool to create module key from container-based software.
  - Using the software in containers is as easy as:

```
$ module load busco
$ busco --version
  BUSCO 5.6.1
```