Magic Tools To Install and Manage Software

Siva Prasad Kasetti

HPC User Services

LSU HPC / LONI

sys-help@loni.org

Louisiana State University

Baton Rouge

March 18, 2026

Magic Tools To Install and Manage Software

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

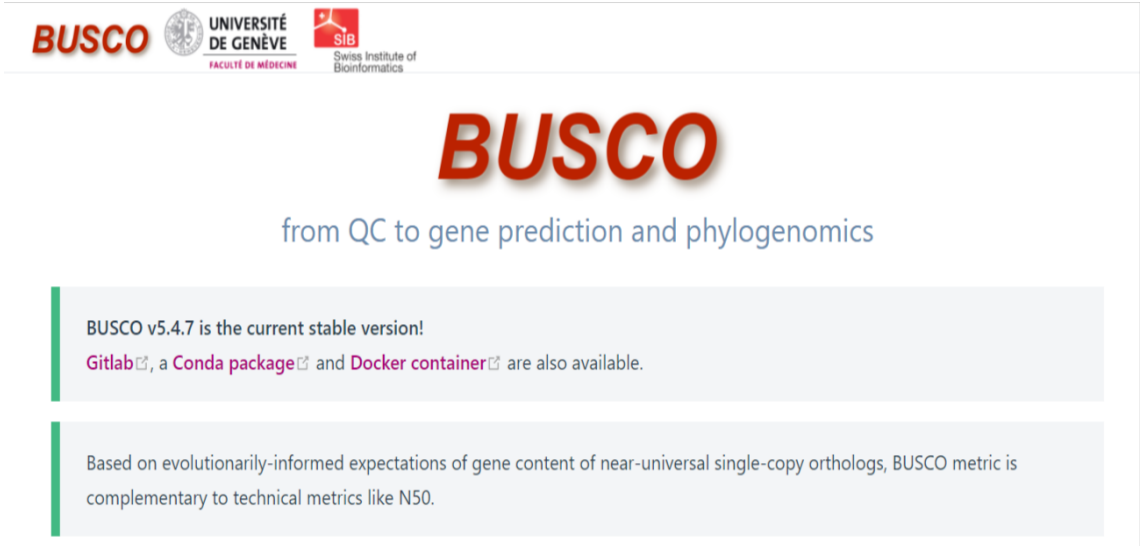
1

Dependencies

Most software depends on other libraries or tools. If those are missing, the software may not install or run.

1) Problems

1. **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.

LSU INFORMATION TECHNOLOGY SERVICES

1) Problems

1. **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

1. **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 … are not available)
  - libboost-iostreams-dev
  - zlib1g-dev
- for comparative AUGUSTUS (multi-species, CGP): (set COMPGENEPRED = false in … 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

1	**Dependencies** Most software depends on other libraries or tools. If those are missing, the software may not install or run.
2	**Version Conflicts** Different packages may need different versions of the same dependency, causing conflicts in a shared environment.

1) Problems

2) Conflicted packages

- Many scientific and ML applications require specific versions of Python, CUDA and libraries like TensorFlow, or PyTorch.
- For example:
  - TensorFlow 2.10 supports Python 3.7–3.10, not Python 3.11,
  - Combine with newer software in a working environment creates version conflicts.

1) Problems

1

Dependencies

Most software depends on other libraries or tools. If those are missing, the software may not install or run.

2

Version Conflicts

Different packages may need different versions of the same dependency, causing conflicts in a shared environment.

3

root permission

Installing software often requires admin access, which most HPC users do not have.

1) Problems

3) **Permission denied** (Welcome to HPC!)

```
[kasetti@mike003 E3RC]$ module load python
[kasetti@mike003 E3RC]$ pip install gdal
```

1) Problems

3) **Permission denied** (Welcome to HPC!)

```
[kasetti@mike003 E3RC]$ module load python
[kasetti@mike003 E3RC]$ pip install gdal
```

```
  writing dependency_links to gdal-utils/GDAL.egg-info/dependency_links.txt
  writing entry points to gdal-utils/GDAL.egg-info/entry_points.txt
  writing requirements to gdal-utils/GDAL.egg-info/requires.txt
  writing top-level names to gdal-utils/GDAL.egg-info/top_level.txt
  Traceback (most recent call last):
    File "<string>", line 92, in fetch_config
    File "/usr/local/packages/python/3.9.7-anaconda/lib/python3.9/subprocess.py", line 951, in __init__
      self._execute_child(args, executable, preexec_fn, close_fds,
    File "/usr/local/packages/python/3.9.7-anaconda/lib/python3.9/subprocess.py", line 1821, in _execute_chi
ld
      raise child_exception_type(errno_num, err_msg, err_filename)
  FileNotFoundError: [Errno 2] No such file or directory: 'gdal-config'
```

1) Problems

3) **Permission denied** (Welcome to HPC!)

```
[kasetti@mike003 E3RC]$ module load python
[kasetti@mike003 E3RC]$ pip install gdal
```

```
  writing dependency_links to gdal-utils/GDAL.egg-info/dependency_links.txt
  writing entry points to gdal-utils/GDAL.egg-info/entry_points.txt
  writing requirements to gdal-utils/GDAL.egg-info/requires.txt
  writing top-level names to gdal-utils/GDAL.egg-info/top_level.txt
  Traceback (most recent call last):
    File "<string>", line 92, in fetch_config
    File "/usr/local/packages/python/3.9.7-anaconda/lib/python3.9/subprocess.py", line 951, in __init__
      self._execute_child(args, executable, preexec_fn, close_fds,
    File "/usr/local/packages/python/3.9.7-anaconda/lib/python3.9/subprocess.py", line 1821, in _execute_chi
ld
      raise child_exception_type(errno_num, err_msg, err_filename)
  FileNotFoundError: [Errno 2] No such file or directory: 'gdal-config'
```

libgdal

1) Problems

3) **Permission denied** (Welcome to HPC!)

- If you ask Google / ChatGPT…

```
$ sudo yum install libgdal-devel        # On Red Hat

$ sudo apt-get install libgdal-dev      # On Ubuntu
```

1) Problems

3) **Permission denied** (Welcome to HPC!)

- If you ask Google / ChatGPT…

```
$ sudo yum install libgdal-devel        # On Red Hat

$ sudo apt-get install libgdal-dev      # On Ubuntu
```

1) Problems

3) **Permission denied** (Welcome to HPC!)

- If you ask Google / ChatGPT…

```
$ sudo yum install libgdal-devel        # On Red Hat

$ sudo apt-get install libgdal-dev      # On Ubuntu
```

1) Problems

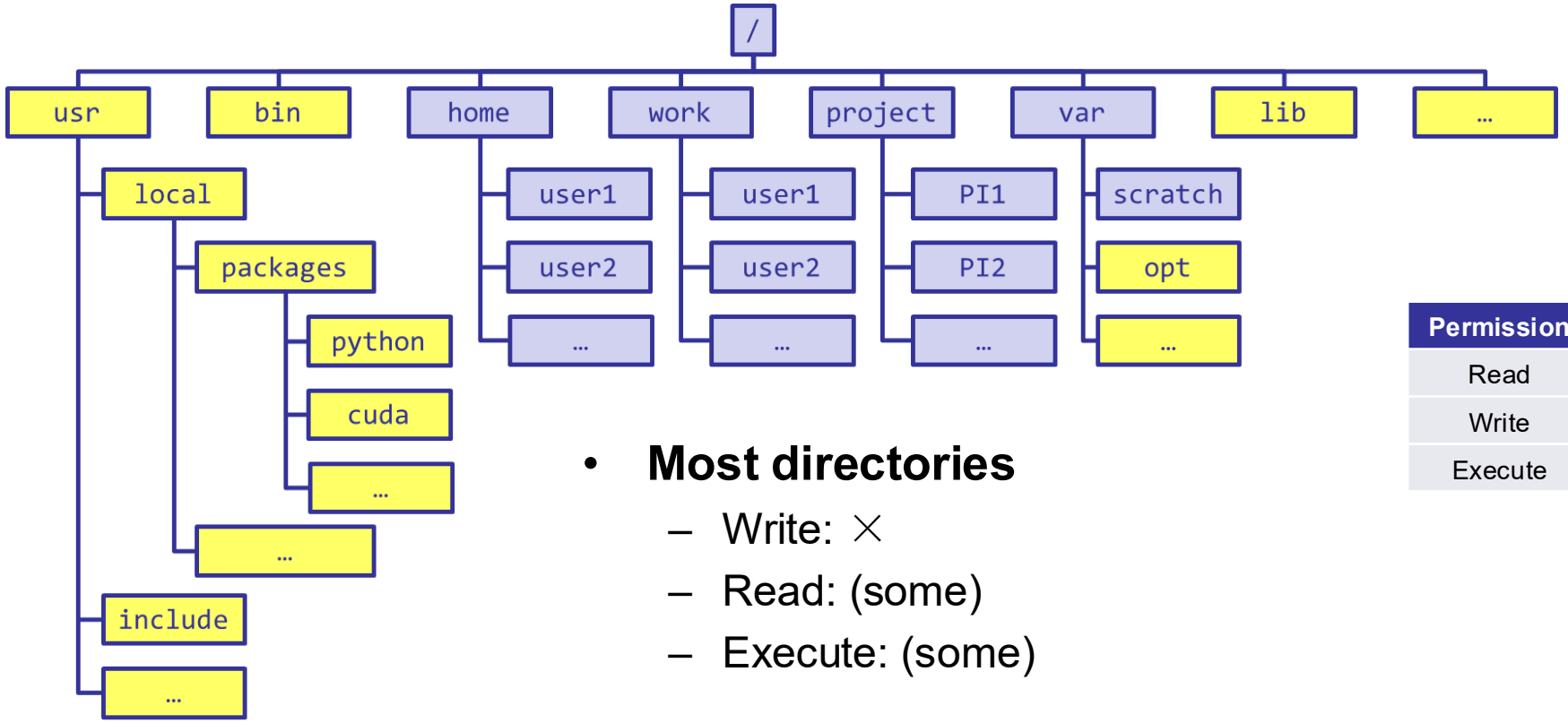
3) Permission denied (Welcome to HPC!)

- If you ask Google / ChatGPT…

```
$ sudo yum install libgdal-devel        # On Red Hat

$ sudo apt-get install libgdal-dev      # On Ubuntu
```

3) **Permission denied** (Welcome to HPC!)



Permission			
Read	√	×	√*
Write	√	×	×
Execute	√	×	√*

- **Most directories**
  - Write: ×
  - Read: (some)
  - Execute: (some)

1) Problems

1

Dependencies

Most software depends on other libraries or tools. If those are missing, the software may not install or run.

2

Version Conflicts

Different packages may need different versions of the same dependency, causing conflicts in a shared environment.

3

root permission

Installing software often requires admin access, which most HPC users do not have.

4

Reproducibility

Reproducing the same setup or sharing it with others can be surprisingly difficult.

4. Reproducibility - Sharing / Migrating your software

– 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

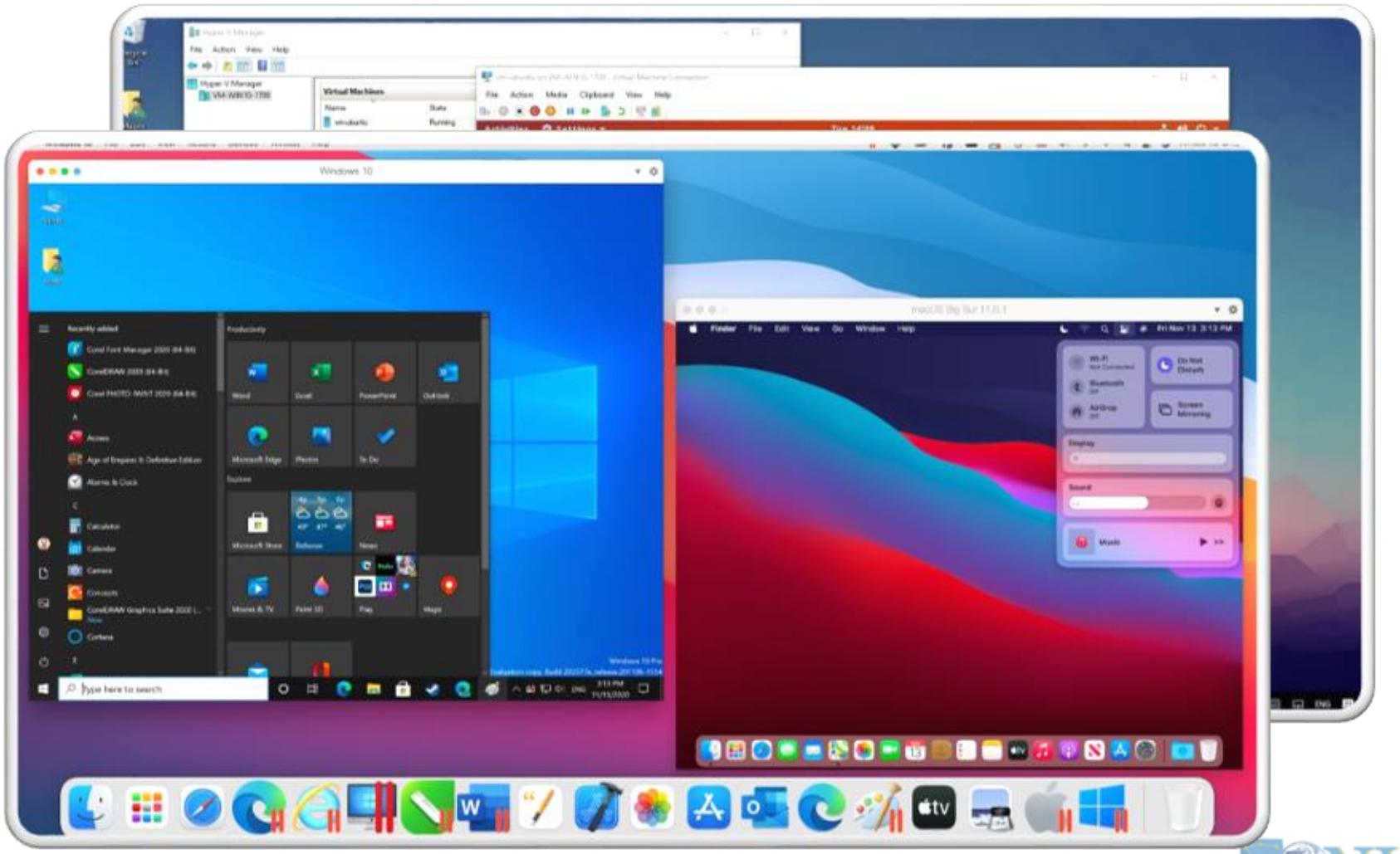
Outlines

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?

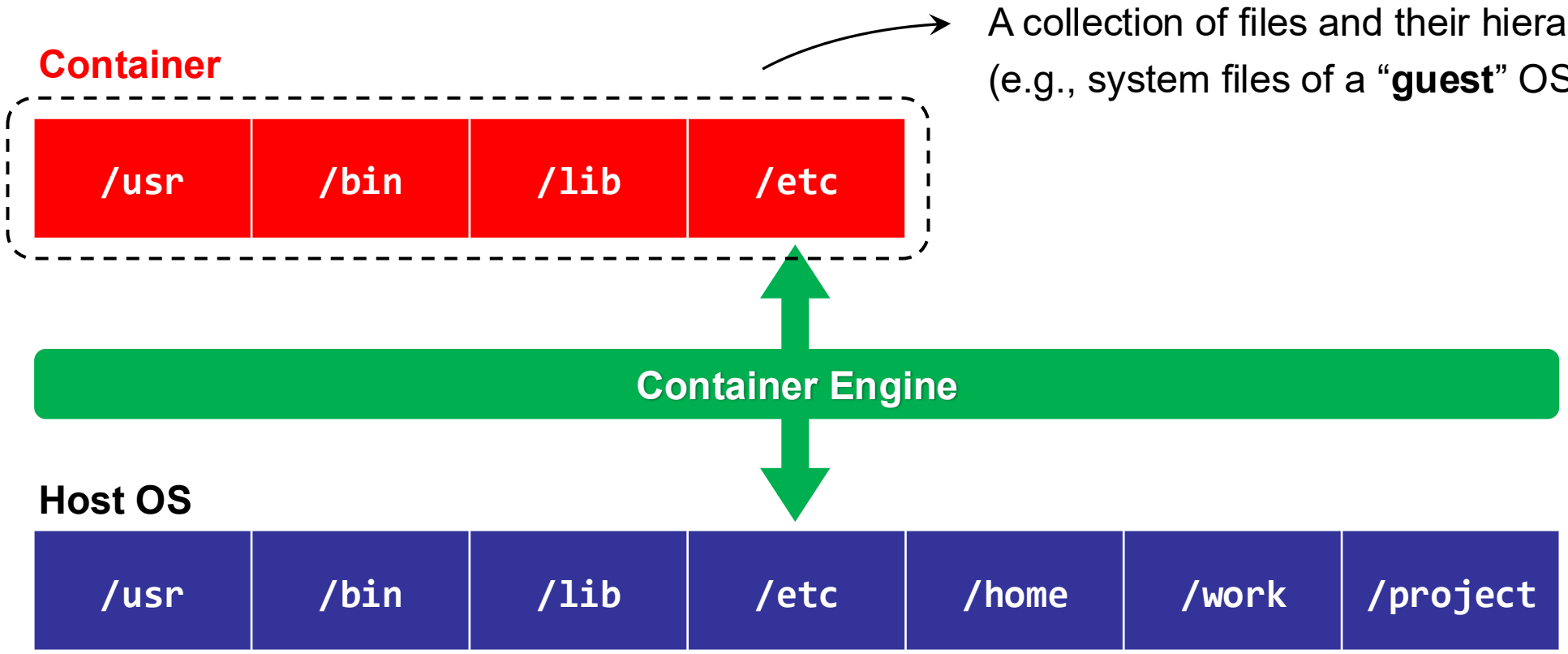
- **Virtual machine**
  - "Virtualize" / "mimic" an **entire computer** on another computer
  - Virtualize both **hardware** and **software**



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**

a) What is a container?



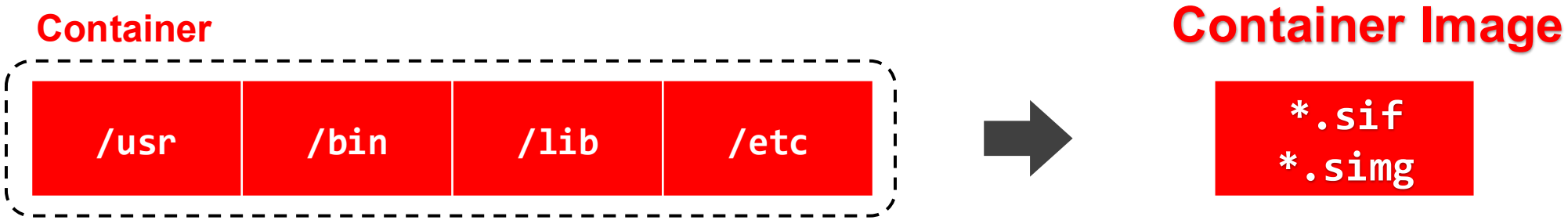
a) What is a container?

/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?

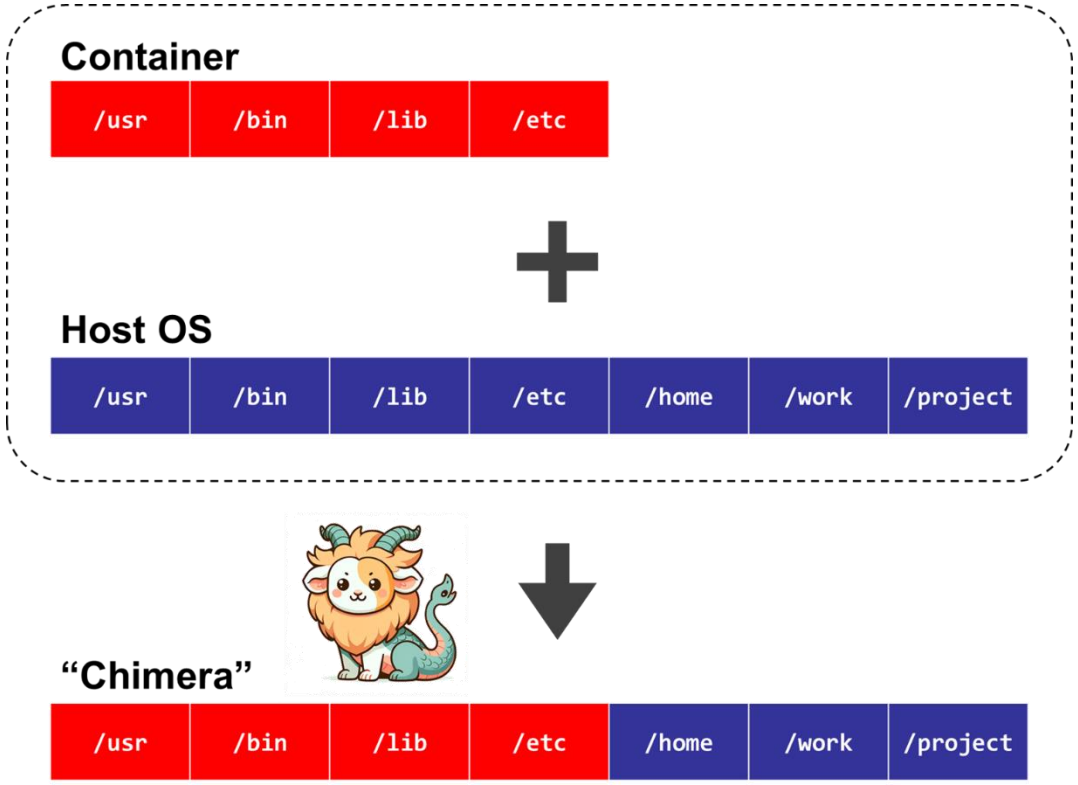


a) What is a container?

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

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

    Whatever happens in a container stays in that container…



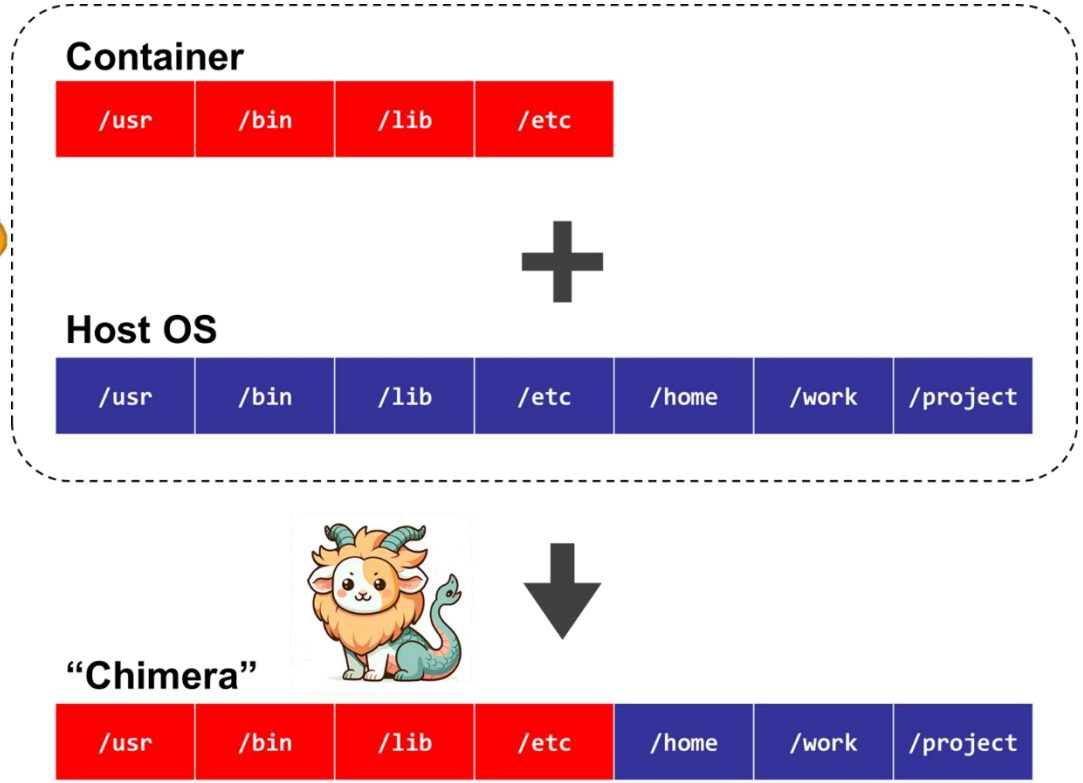
b) How does it solve my problems?

- **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

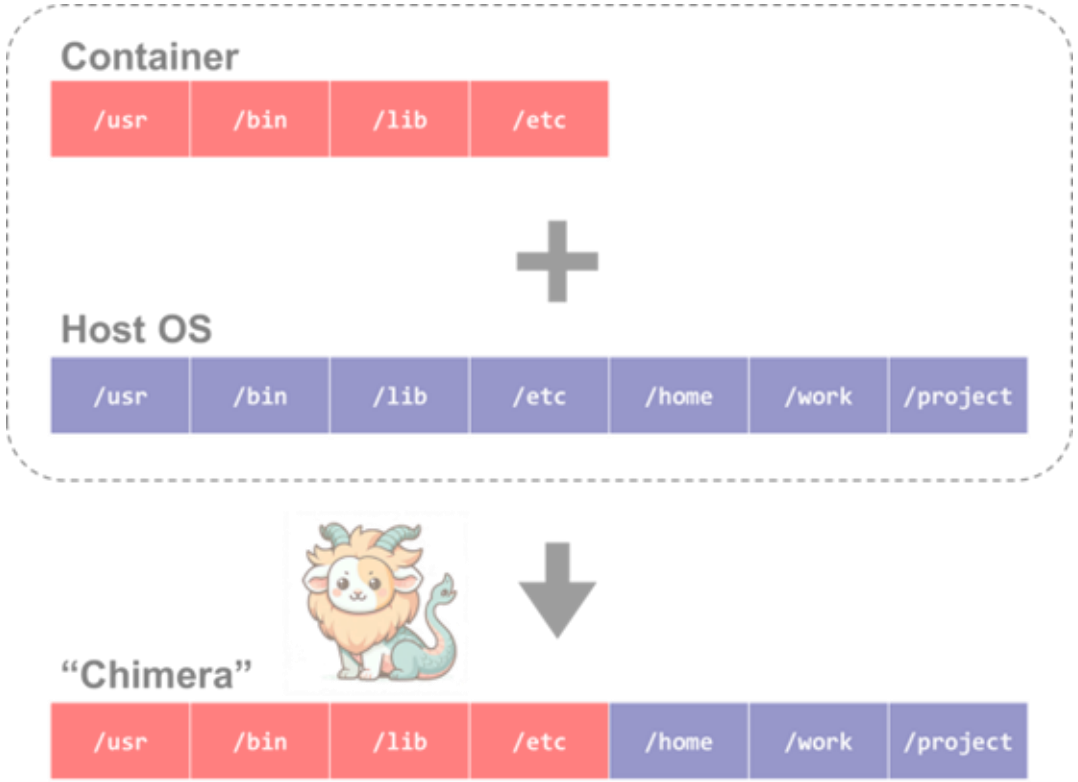
- **Conflicted packages**
  - Install in different containers.

- **Share / Migrate**
  - Copy-paste a container image!



c) What is Singularity?

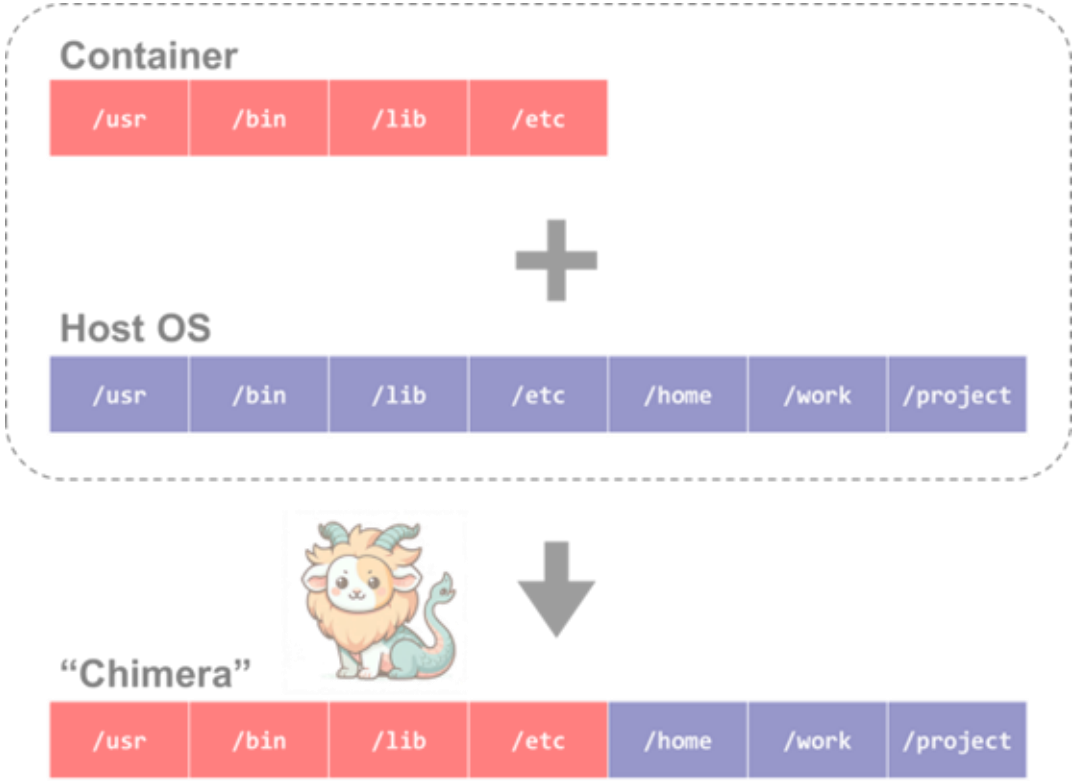
Technology →



c) What is Singularity?

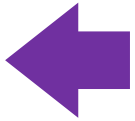
Singularity

↑ **Software** system that implements the technology



c) What is Singularity?

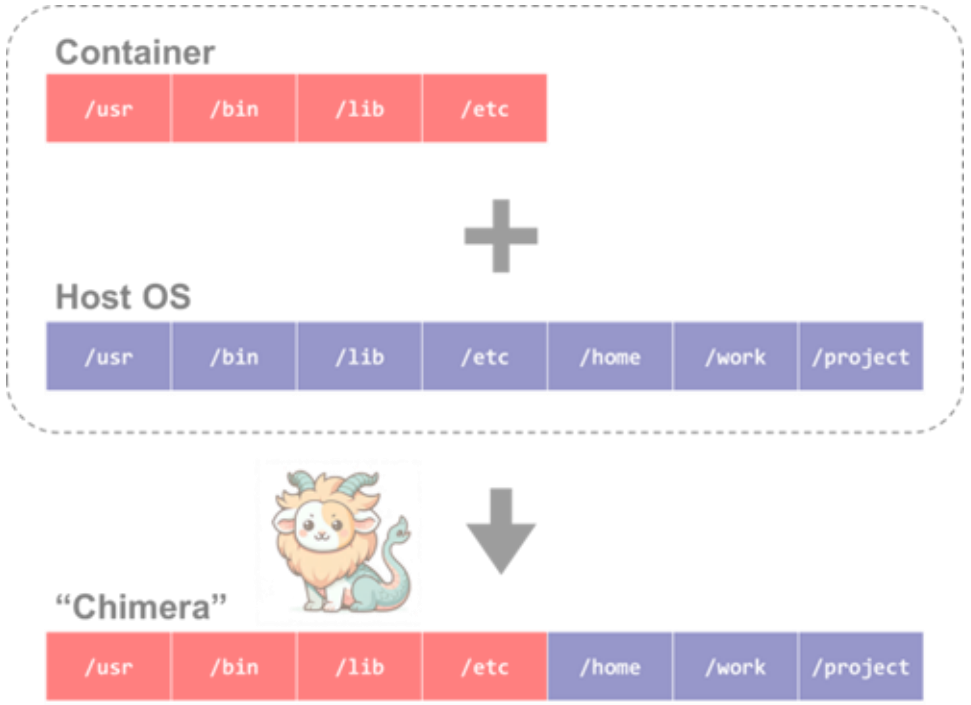
c) What is Singularity?



- 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

Basic Information

- **Singularity availability**

a) On **all clusters**

- ✓ **LSU HPC**: SMIC, Deep Bayou, SuperMike 3
- ✓ **LONI**: QB3, QB4

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

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) Basic commands

- **Available images**

  – On all clusters: `/project/containers/images`

```
[kasetti@qbd1 ~]$ ls /project/containers/images/
adcirc.5bc04d6.centos8.sif                  mesos_1.9.0_ubuntu16.04.simg
adcirc.5bc04d6.rky8.5.sif                   minknow22.sif
adcirc.5bc04d6.sif                          nco-4.9.8.sif
agat-1.4.0.sif                              nextflow-23.10.1.sif
alphafold-catgumag-2.2.sif                  nvidia-dli.sif
alps-2.3.0-dockerhub.simg                   octopus-14.0.sif
alps-2.3.0-dockerhub-v2.simg                ollamaondemand-1.0.0.sif
bcftools-1.18.sif                           ollamaondemand-1.0.0.sif.old
beast2-2.7.7.sif                            openbabel-3.1.1.sif
blast-2.14.1.sif                            openfoam10-openmpi.4.0.3-pmi2.sif
blender-2.79b-cuda-8.0-ubuntu-16.04.simg    openfoam9.sdfibm-openmpi.4.0.3-pmi2.sif
bowtie2-2.5.1.sif                           openfoam-v2212-openmpi.4.1.2-pmi2.sif
braker-3.0.8.sif                            openfoam.v9.sdfibm.sif
busco-5.7.1.sif                             paraviewweb-5.12.1.sif
```

2) Basic commands

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

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

Try me: `/project/containers/images/singularity-training-ubuntu.sif`

2) Basic commands

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

Syntax		Description
`singularity shell [options] <container>`		Starts an interactive shell in the image
`[Options]`	`-B /path/to/bind`	Bind a path(s) • Access host files with bind mounts • /home, /work 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: `/project/containers/images/singularity-training-ubuntu.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

2) Basic commands

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>`	Starts an interactive shell in the image
`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

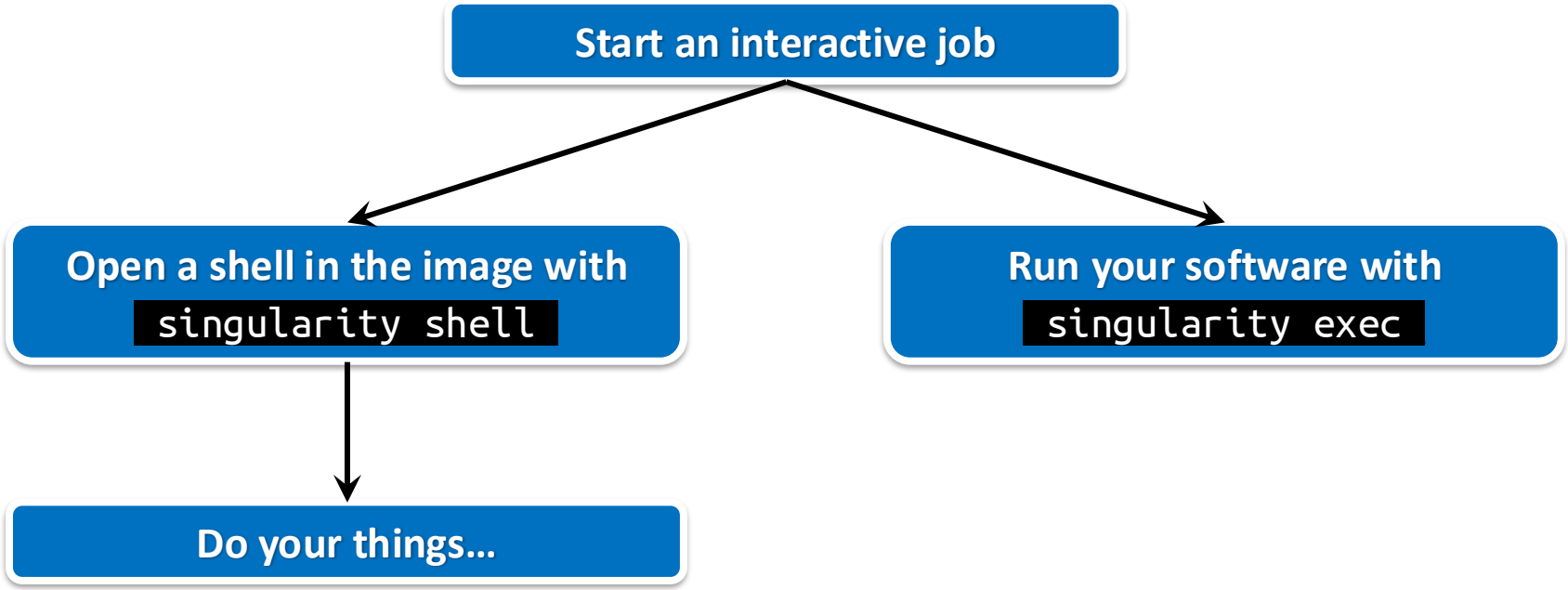
3) Run jobs with Singularity

- **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

```
#!/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/singularity-
training-ubuntu.sif

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

Example

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: `/project/containers/images`

```
[kasetti@qbd1 ~]$ ls /project/containers/images/
adcirc.5bc04d6.centos8.sif                  mesos_1.9.0_ubuntu16.04.simg
adcirc.5bc04d6.rky8.5.sif                   minknow22.sif
adcirc.5bc04d6.sif                          nco-4.9.8.sif
agat-1.4.0.sif                              nextflow-23.10.1.sif
alphafold-catgumag-2.2.sif                  nvidia-dli.sif
alps-2.3.0-dockerhub.simg                   octopus-14.0.sif
alps-2.3.0-dockerhub-v2.simg                ollamaondemand-1.0.0.sif
bcftools-1.18.sif                           ollamaondemand-1.0.0.sif.old
beast2-2.7.7.sif                            openbabel-3.1.1.sif
blast-2.14.1.sif                            openfoam10-openmpi.4.0.3-pmi2.sif
blender-2.79b-cuda-8.0-ubuntu-16.04.simg    openfoam9.sdfibm-openmpi.4.0.3-pmi2.sif
bowtie2-2.5.1.sif                           openfoam-v2212-openmpi.4.1.2-pmi2.sif
braker-3.0.8.sif                            openfoam.v9.sdfibm.sif
busco-5.7.1.sif                             paraviewweb-5.12.1.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

BEFORE

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

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

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 !**

techradar pro THE BUSINESS TECHNOLOGY EXPERTS US Edition

News Reviews Features Expert Insights Website builders Web hosting

TRENDING Expert Insights Best web hosting Best website builder

When you purchase through links on our site, we may earn an affiliate commission. Here's how it works.

Pro > Security

Malware attacks on Docker Hub spread millions of malicious repositories

News By Sead Fadilpašić published yesterday

Millions of repositories contained nothing but links to malicious websites

[1] https://www.techradar.com/pro/security/malware-attacks-on-docker-hub-spread-millions-of-malicious-repositories

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/ • **BUT!** Do **NOT** just trust everything you see there! • Look for trustworthy icons like 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

- **BONUS: Hot packages!**

  i. **PyTorch** (w/ GPU support)

```
$ singularity pull docker://pytorch/pytorch:2.10.0-cuda13.0-cudnn9-devel
```

  ii. **Tensorflow** (w/ GPU support)

```
$ singularity pull docker://tensorflow/tensorflow:2.21.0-gpu-jupyter
```

3) How to get

- **BONUS: Hot packages!**

  i. **PyTorch** (w/ GPU support)

```
$ module load pytorch
```

  ii. **Tensorflow** (w/ GPU support)

```
$ module load tensorflow
```

Summary

- **Pull the image or/and build the image**

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

Outlines

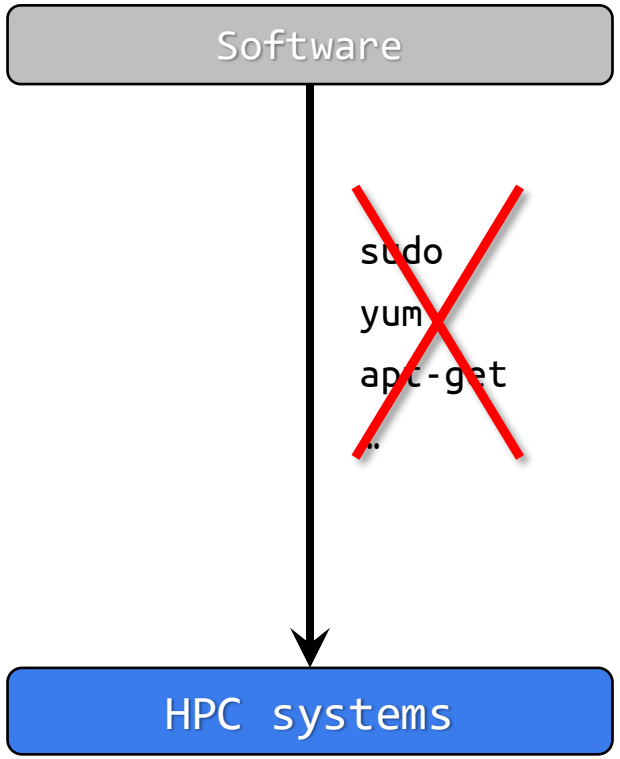
1. Why Container?
   1) Problems
   2) Container & Singularity
2. Run an Existing Container Image
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity
3. Get More Container Images
   1) What you need
   2) Where to get
   3) How to get
4. **Build Your Own Container Image**
  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

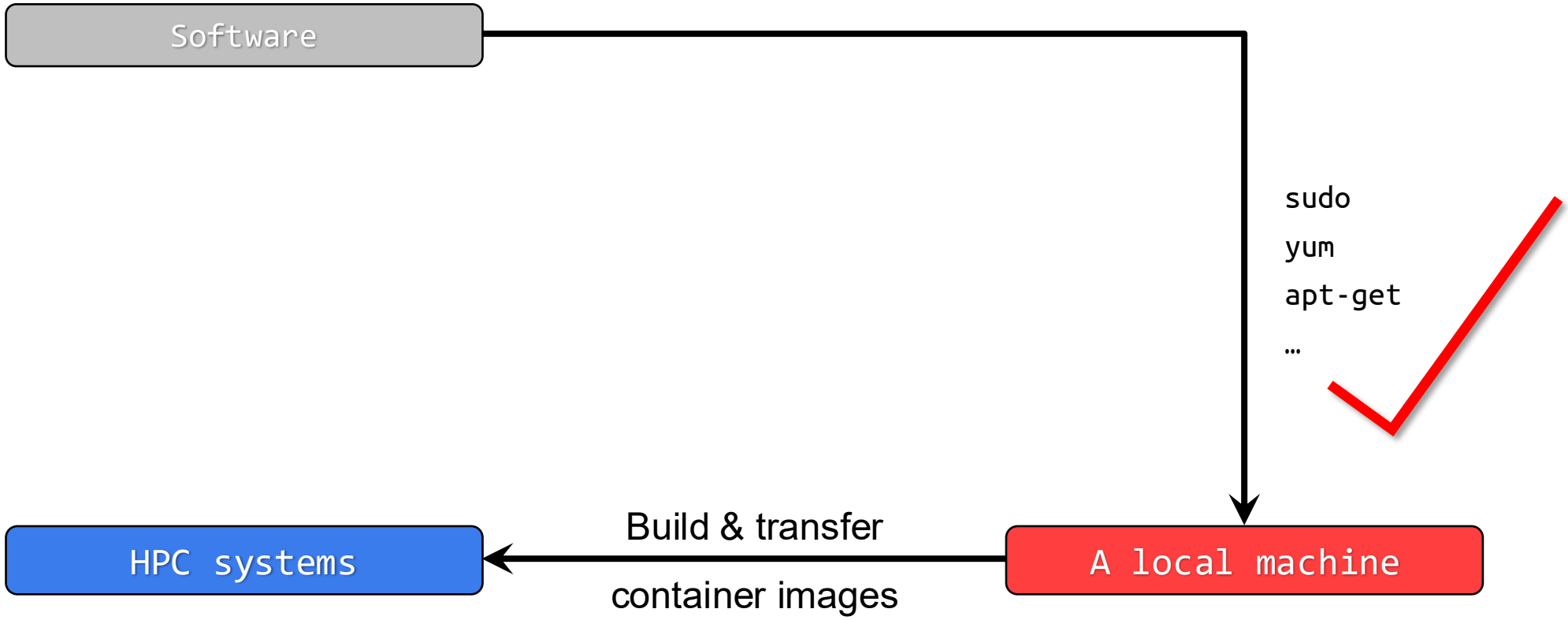
4. Build Your Own Container Image

- **Idea**



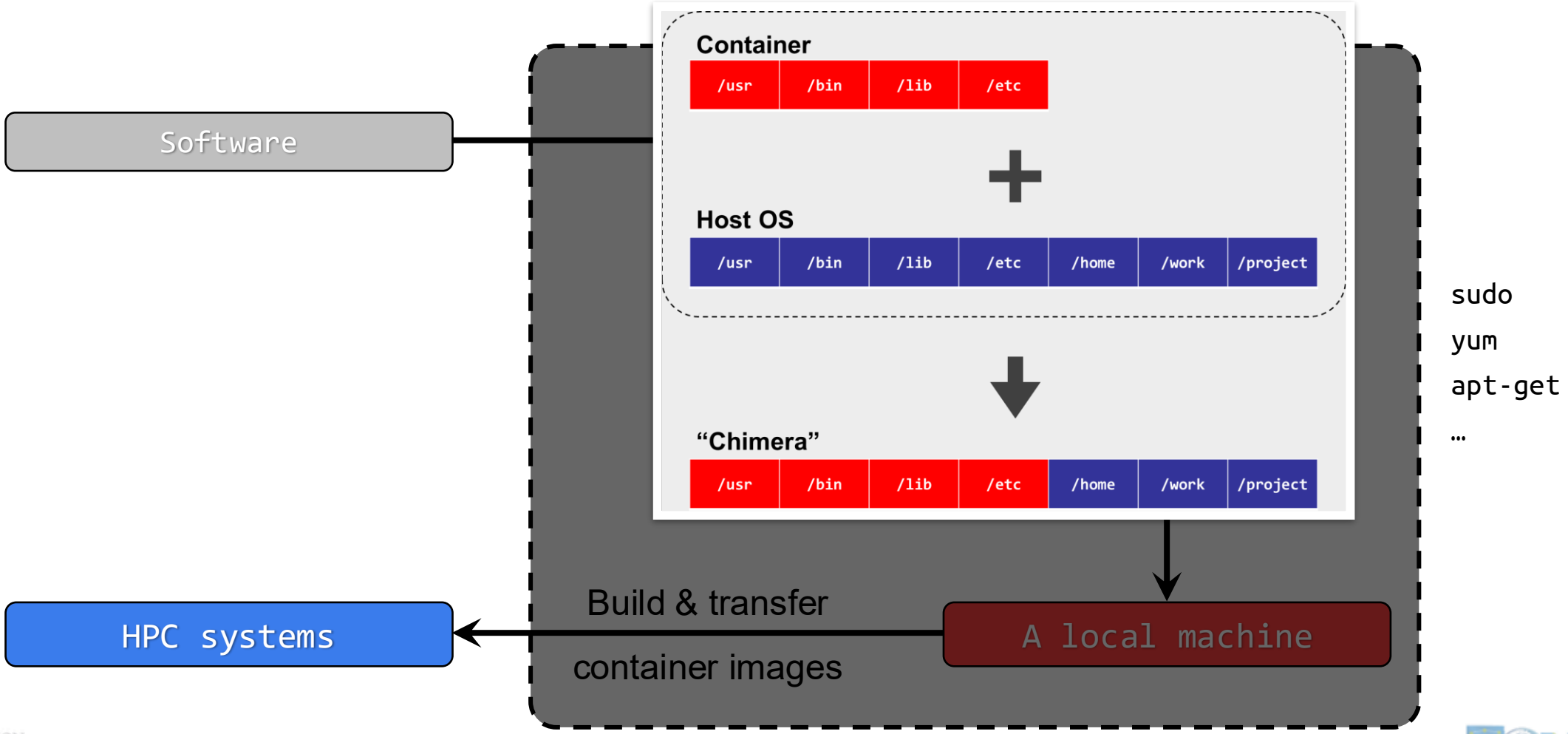
4. Build Your Own Container Image

- **Idea**



4. Build Your Own Container Image

- **Idea**



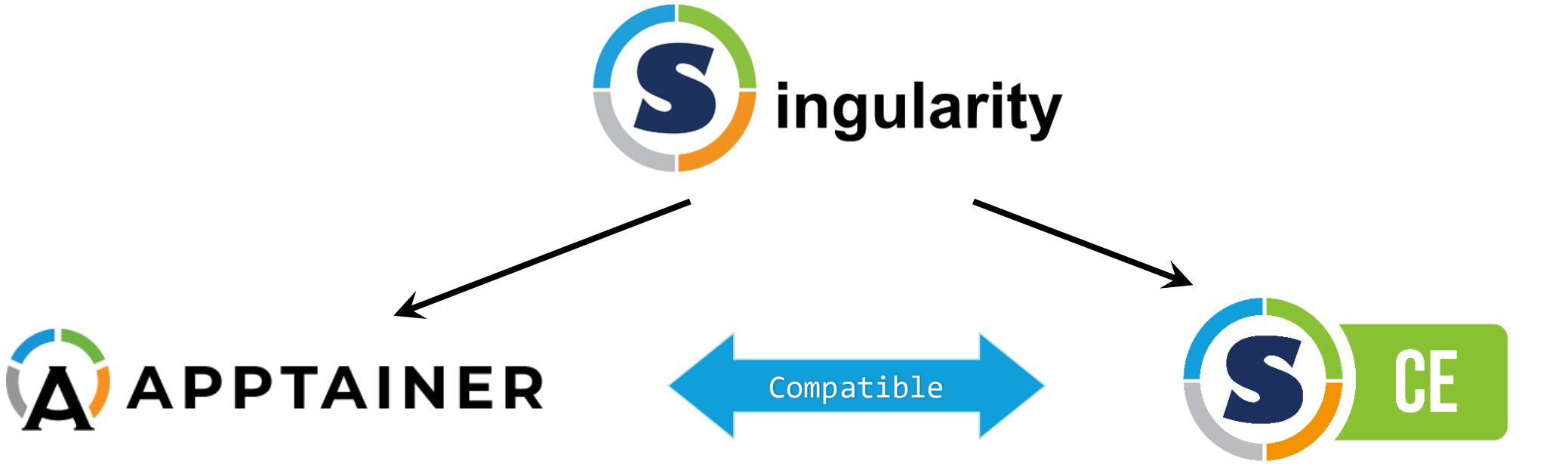
Outlines

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**

- **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

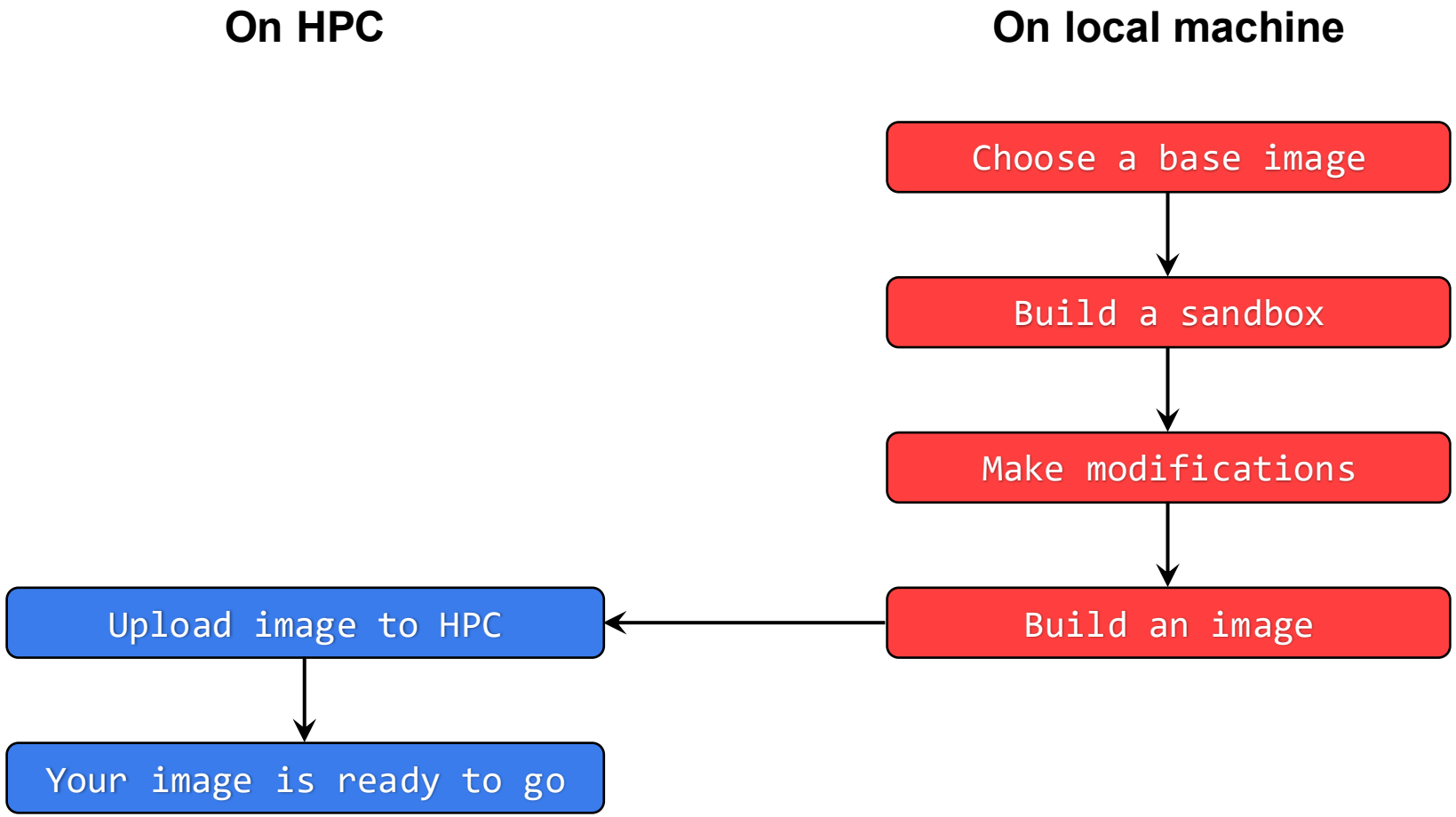
1. Why Container?
   1) Problems
   2) Container & Singularity

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

   – Read-only without it

2) Typical workflow

c) Make modifications

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

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

- Read-only without it

2) Typical workflow

c) Make modifications

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

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

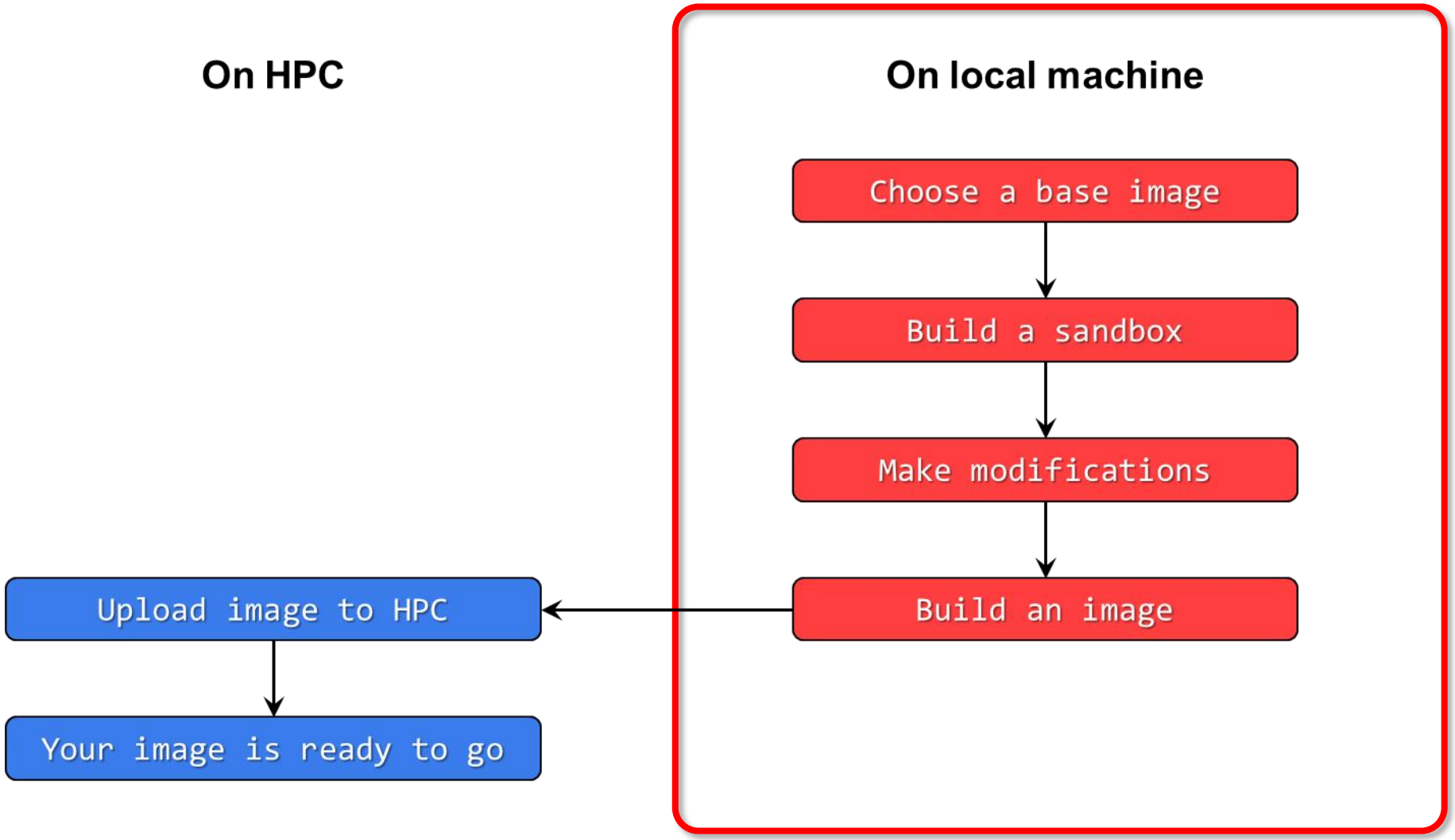
- **Quick recap**

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 ...`

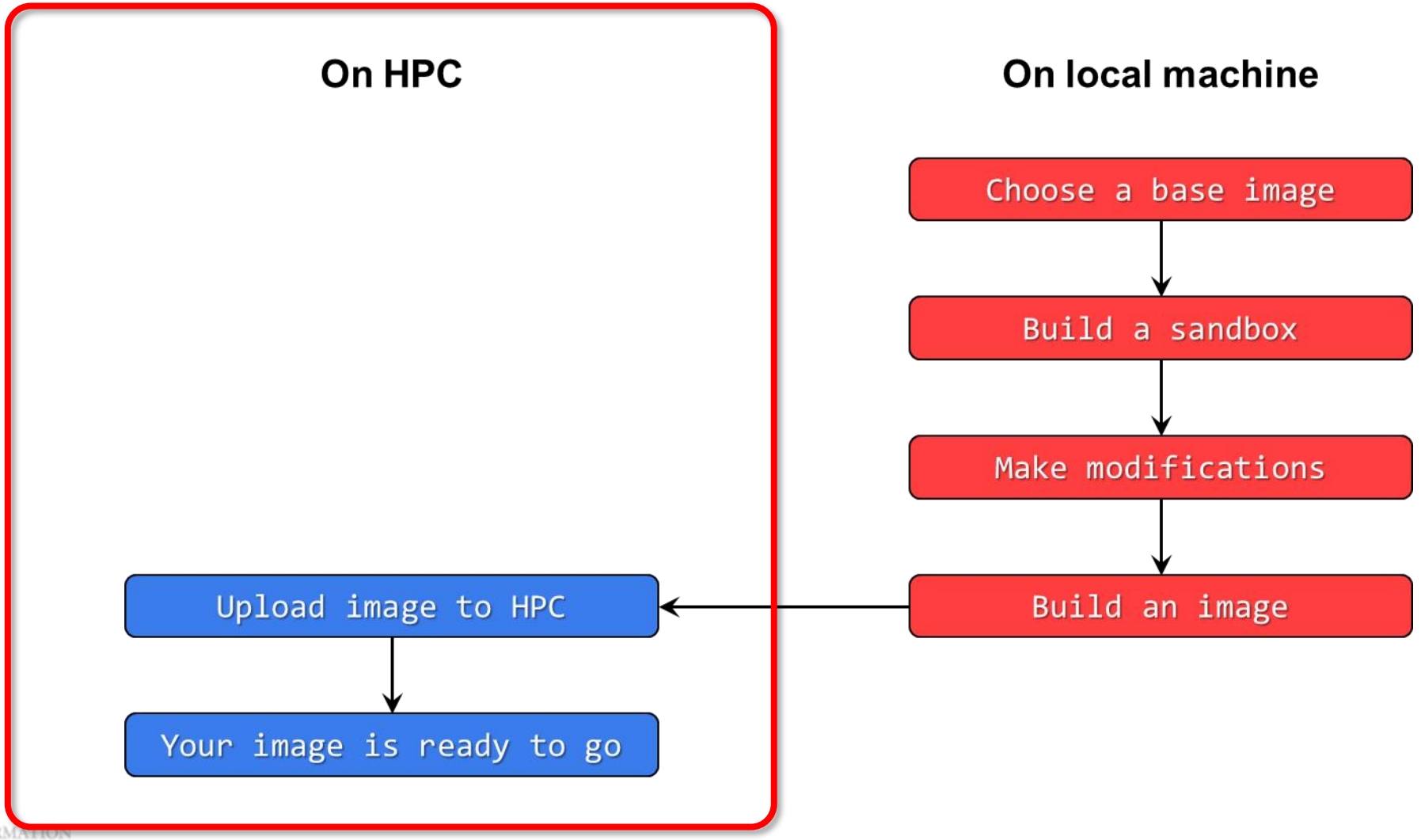
e) Upload image to HPC and run

Now! The moment of truth!

2) Typical workflow



2) Typical workflow

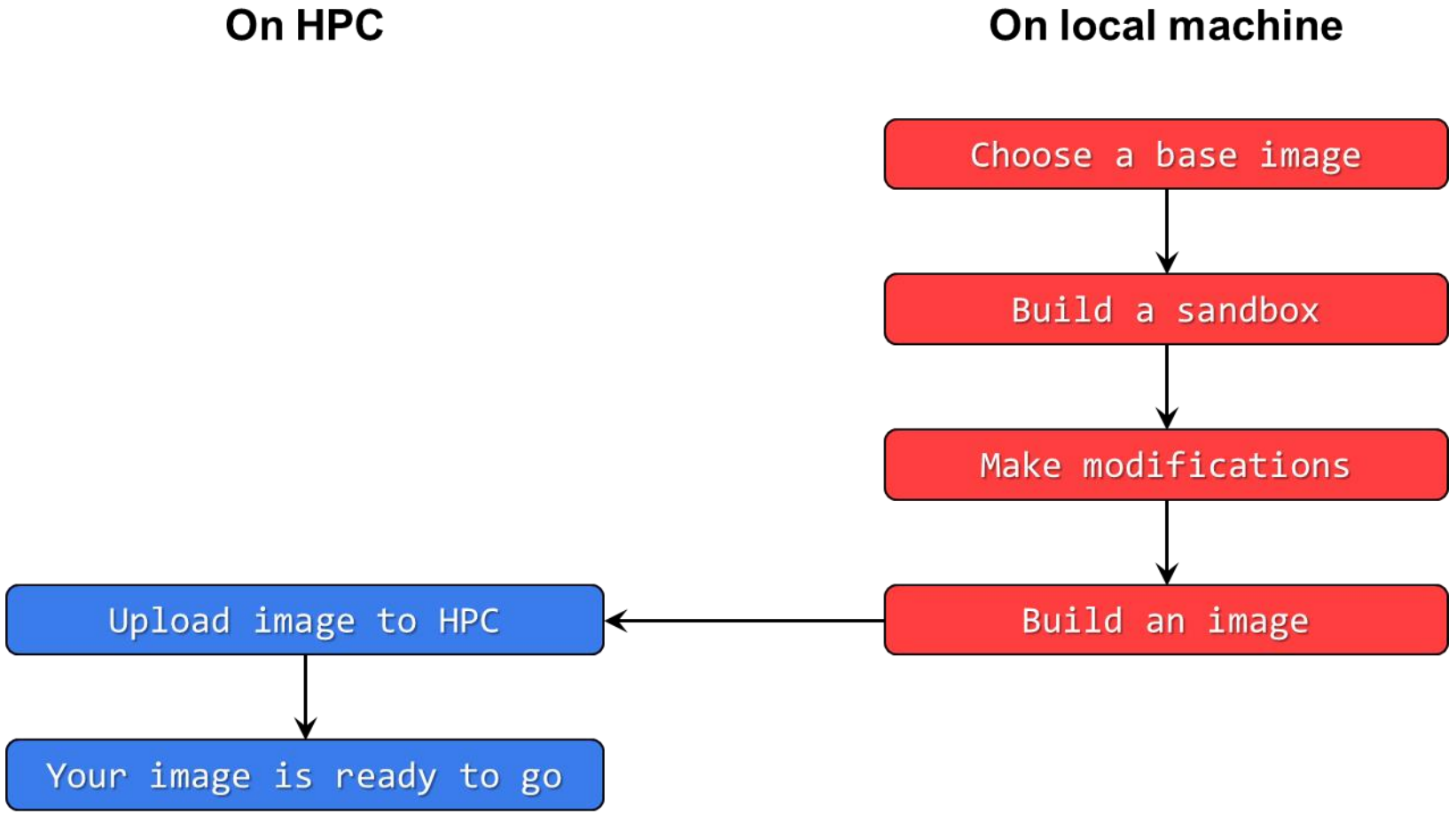


Outlines

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

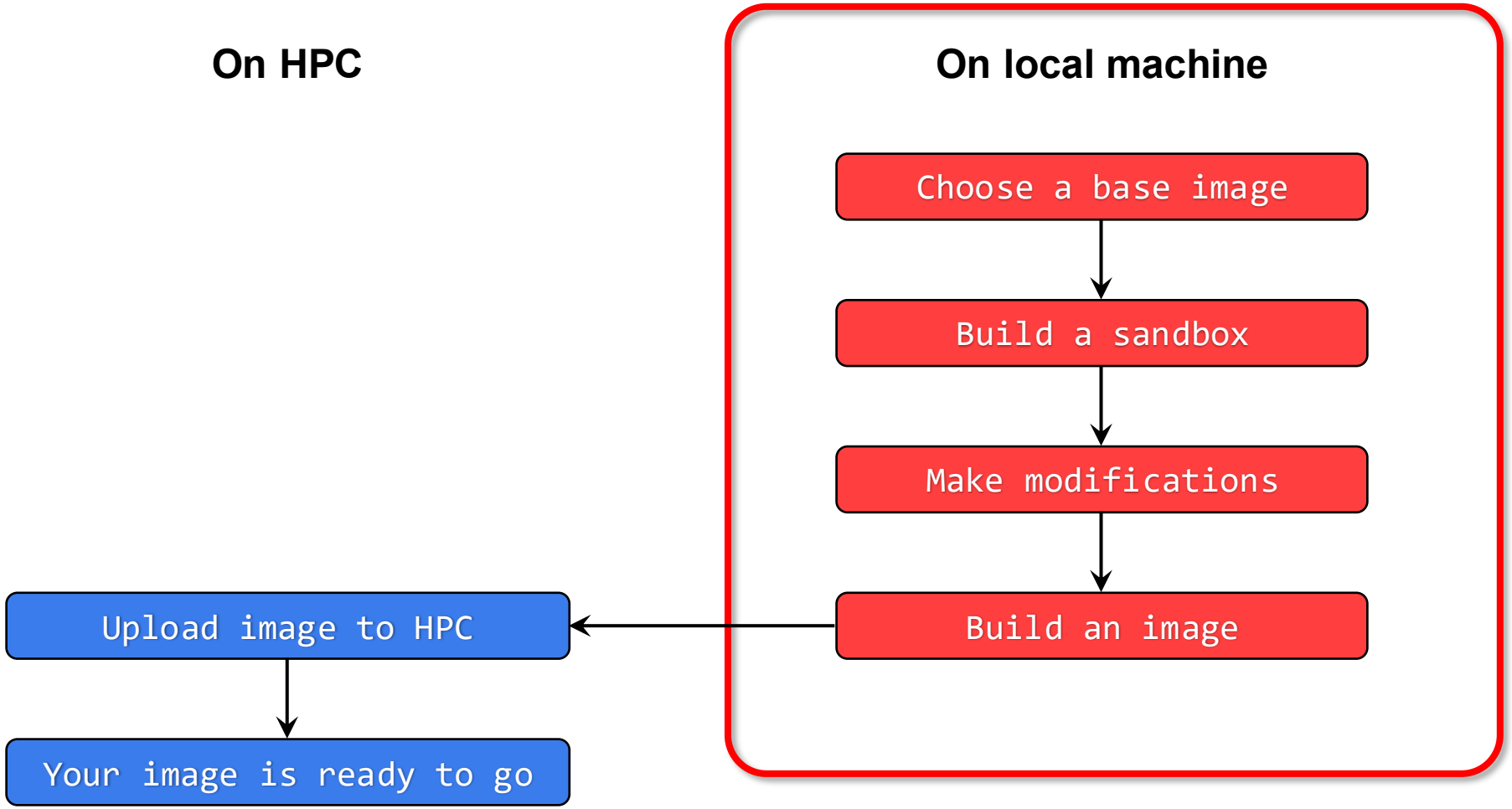
- **Why?**

Pros	Cons
• Flexibility	• Repeatability • Minimizing image size

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

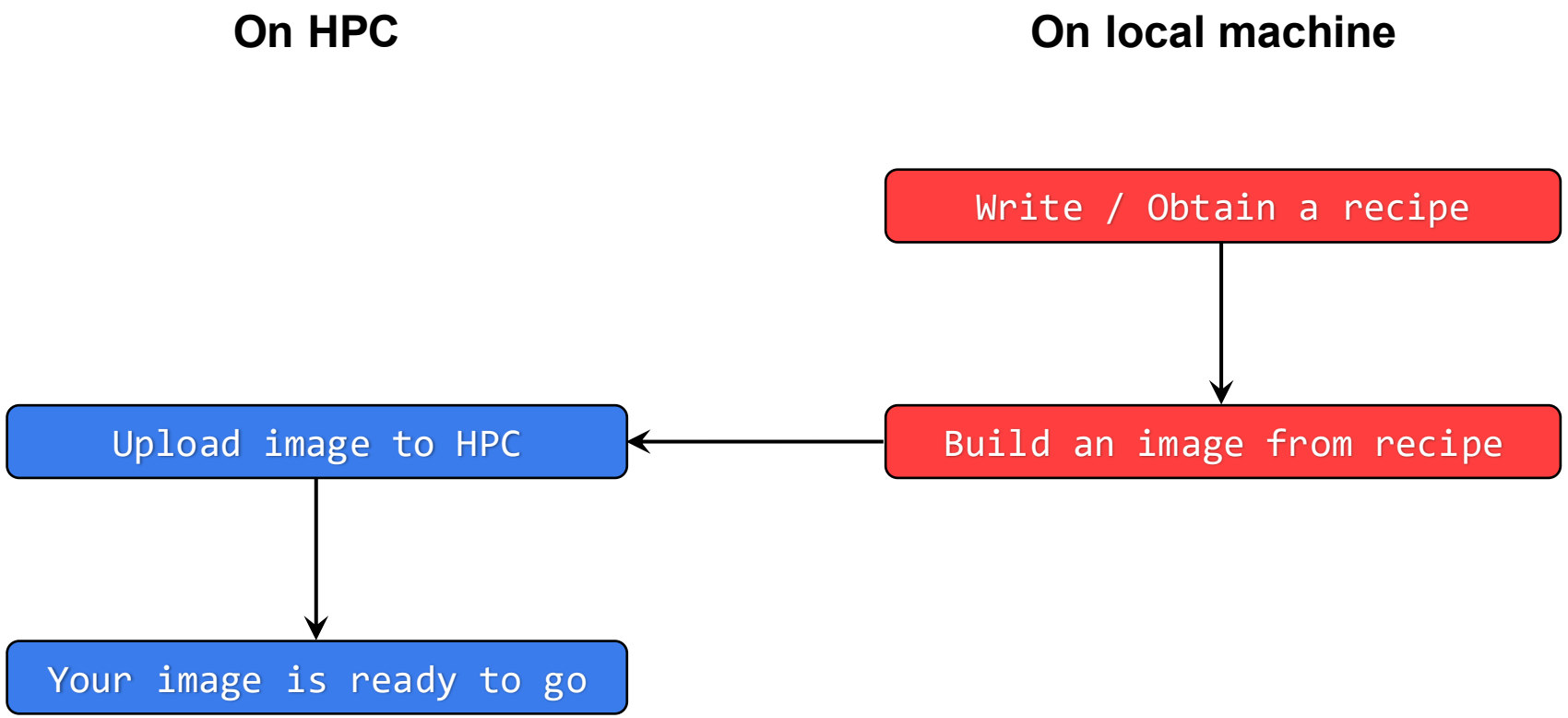
3) Make it easier - Recipe

- **Why?**



3) Make it easier - Recipe

- **Why?**



3) Make it easier - Recipe

a) What does a recipe look like?

recipe.def

```
Bootstrap: docker
From: ubuntu:24.04

%labels
    Author Siva Prasad Kasetti
    Version 1.0
    Description Install curl

%post
    apt-get update -y
    apt-get install -y curl

%environment
   export MYENV="This is my environment variable."

%runscript
   echo "Print curl verson:"
   curl --version
```

3) Make it easier - Recipe

a) What does a recipe look like?

```
Bootstrap: docker
From: ubuntu:24.04

%labels
    Author Siva Prasad Kasetti
    Version 1.0
    Description Install curl

%post
    apt-get update -y
    apt-get install -y curl

%environment
   export MYENV="This is my environment variable."

%runscript
   echo "Print curl verson:"
   curl --version
```

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

3) Make it easier - Recipe

a) What does a recipe look like?

```
Bootstrap: docker
From: ubuntu:24.04

%labels
    Author Siva Prasad Kasetti
    Version 1.0
    Description Install curl

%post
    apt-get update -y
    apt-get install -y curl

%environment
   export MYENV="This is my environment variable."

%runscript
   echo "Print curl verson:"
   curl --version
```

Label

- Container information (write whatever you want)

3) Make it easier - Recipe

a) What does a recipe look like?

```
Bootstrap: docker
From: ubuntu:24.04

%labels
    Author Siva Prasad Kasetti
    Version 1.0
    Description Install curl

%post
    apt-get update -y
    apt-get install -y curl

%environment
   export MYENV="This is my environment variable."

%runscript
   echo "Print curl verson:"
   curl --version
```

Post
- Commands to execute after the base image is pulled

3) Make it easier - Recipe

a) What does a recipe look like?

```
Bootstrap: docker
From: ubuntu:24.04

%labels
    Author Siva Prasad Kasetti
    Version 1.0
    Description Install curl

%post
    apt-get update -y
    apt-get install -y curl

%environment
   export MYENV="This is my environment variable."

%runscript
   echo "Print curl verson:"
   curl --version
```

Environment

- Define environmental variables every time the container is executed

3) Make it easier - Recipe

a) What does a recipe look like?

```
Bootstrap: docker
From: ubuntu:24.04

%labels
    Author Siva Prasad Kasetti
    Version 1.0
    Description Install curl

%post
    apt-get update -y
    apt-get install -y curl

%environment
   export MYENV="This is my environment variable."

%runscript
   echo "Print curl verson:"
   curl --version
```

Runscript
- Commands to be run with `singularity run`

3) Make it easier - Recipe

a) What does a recipe look like?

```
Bootstrap: docker
From: ubuntu:24.04

%labels
    Author Siva Prasad Kasetti
    Version 1.0
    Description Install curl

%post
    apt-get update -y
    apt-get install -y curl

%environment
   export MYENV="This is my environment variable."

%runscript
   echo "Print curl verson:"
   curl --version
```

3) Make it easier - Recipe

a) What does a recipe look like?

```
Bootstrap: docker
From: ubuntu:24.04

%labels
    Author Siva Prasad Kasetti
    Version 1.0
    Description Install curl

%post
    apt-get update -y
    apt-get install -y curl

%environment
   export MYENV="This is my environment variable."

%runscript
   echo "Print curl verson:"
   curl --version
```

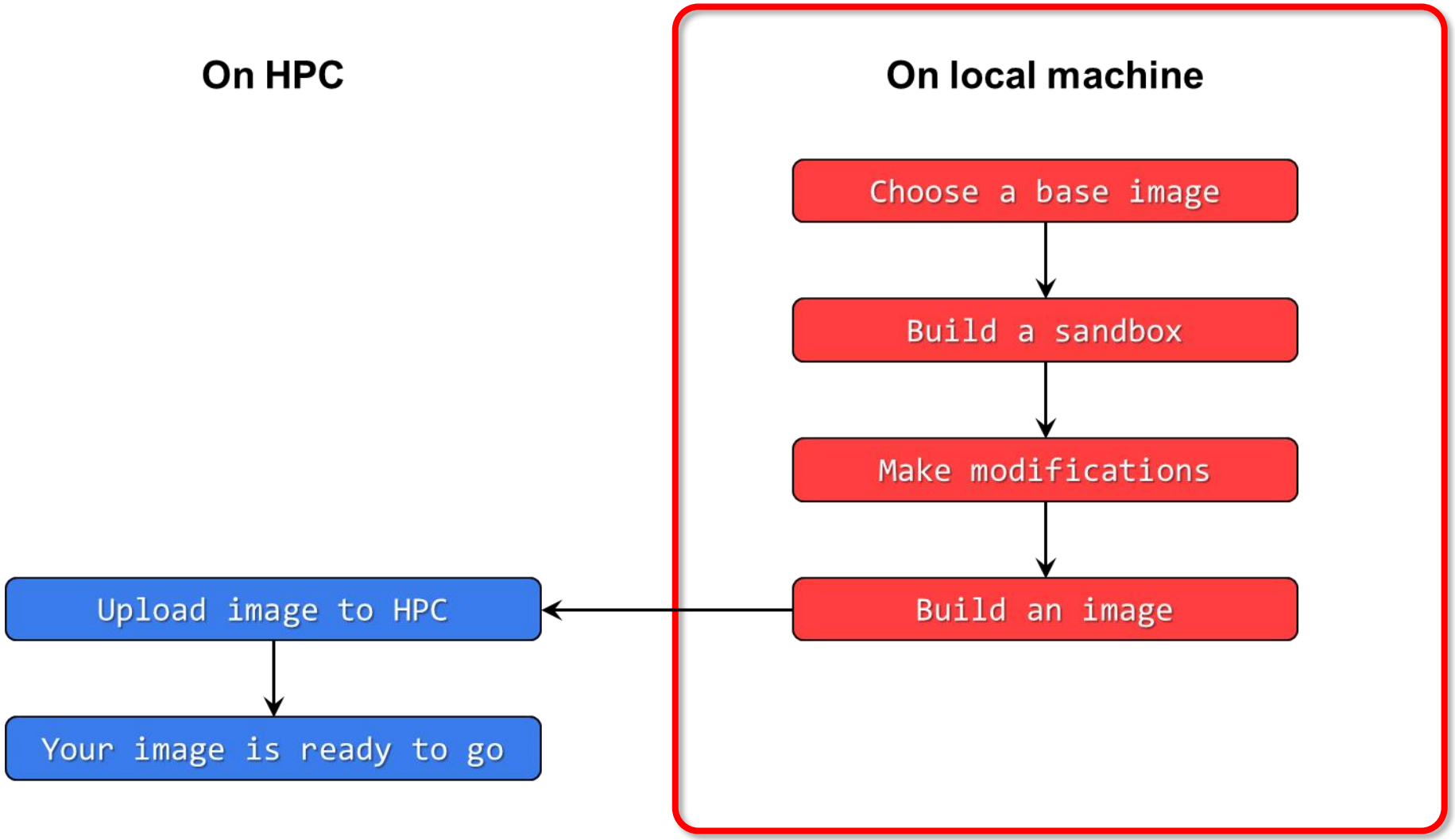
3) Make it easier - Recipe

b) Build the recipe

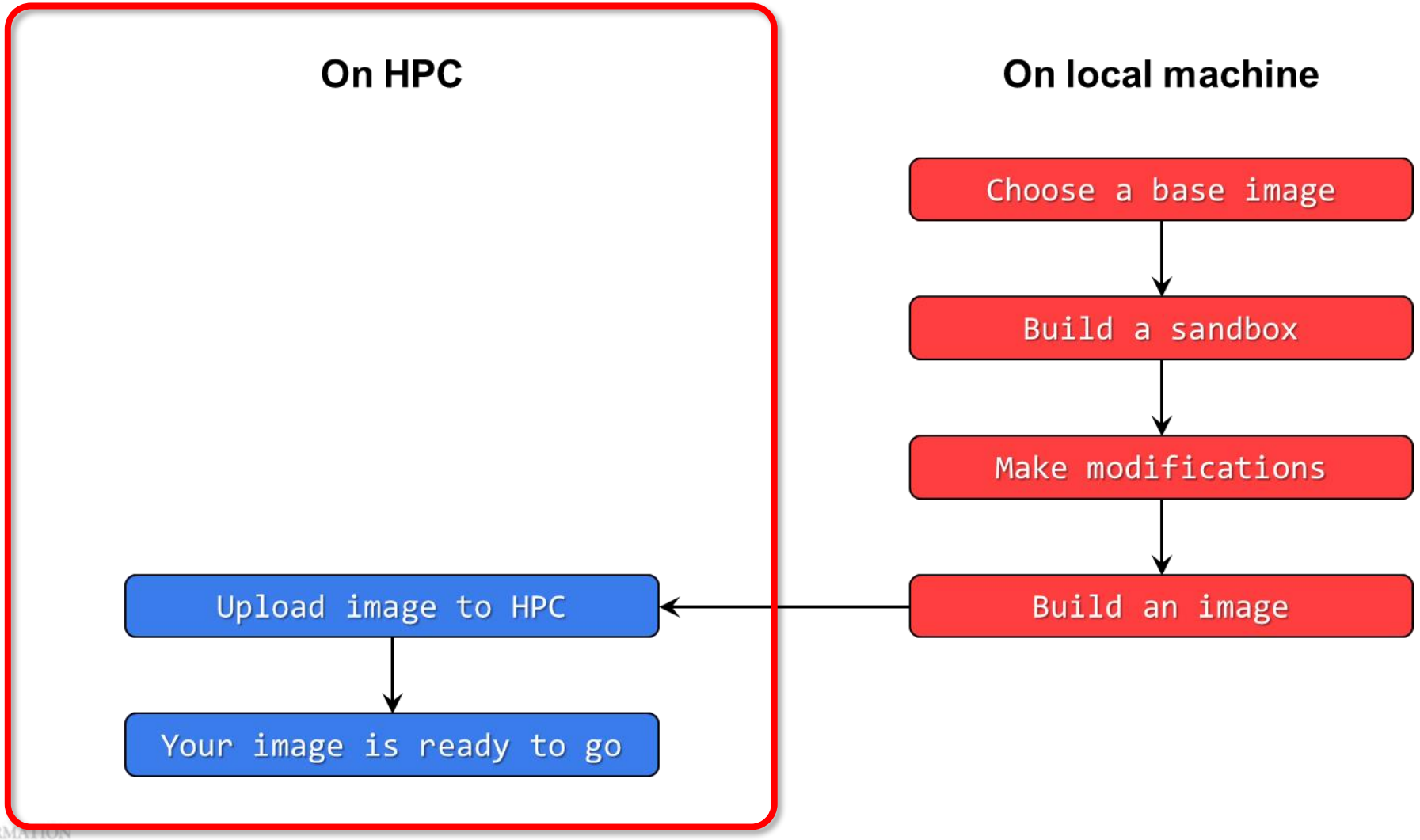
```
$ singularity build [options] <target> <source>
```

<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


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

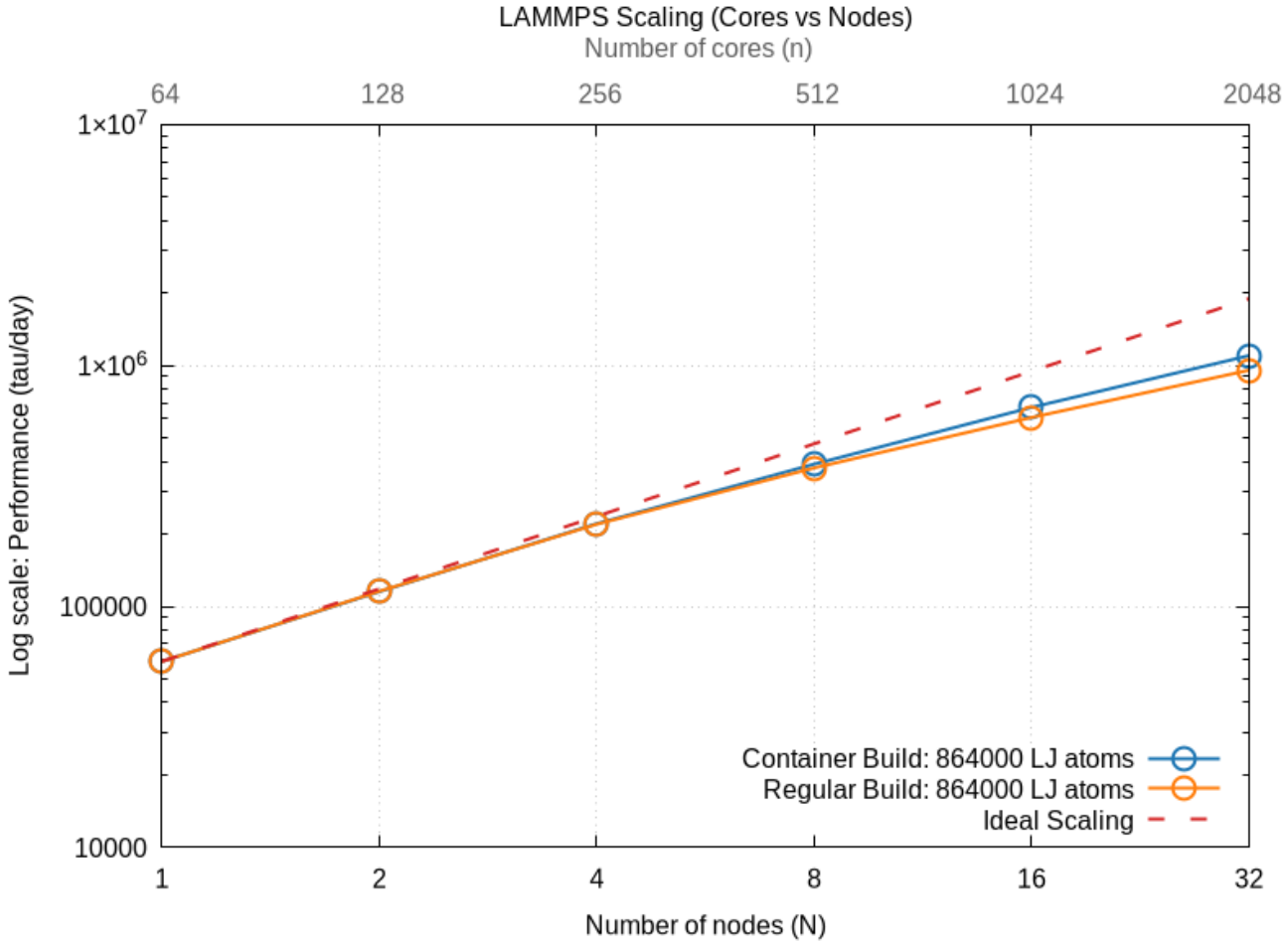
2. **Run an Existing Container Image**
   1) What you need
   2) Basic commands
   3) Running jobs with Singularity

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

Performance Comparison

- The MD simulation package **LAMMPS** was used as the benchmark.
- Performance was compared between running LAMMPS directly from source and running it inside a container.
- Results showed identical performance between both approaches.



Contact us

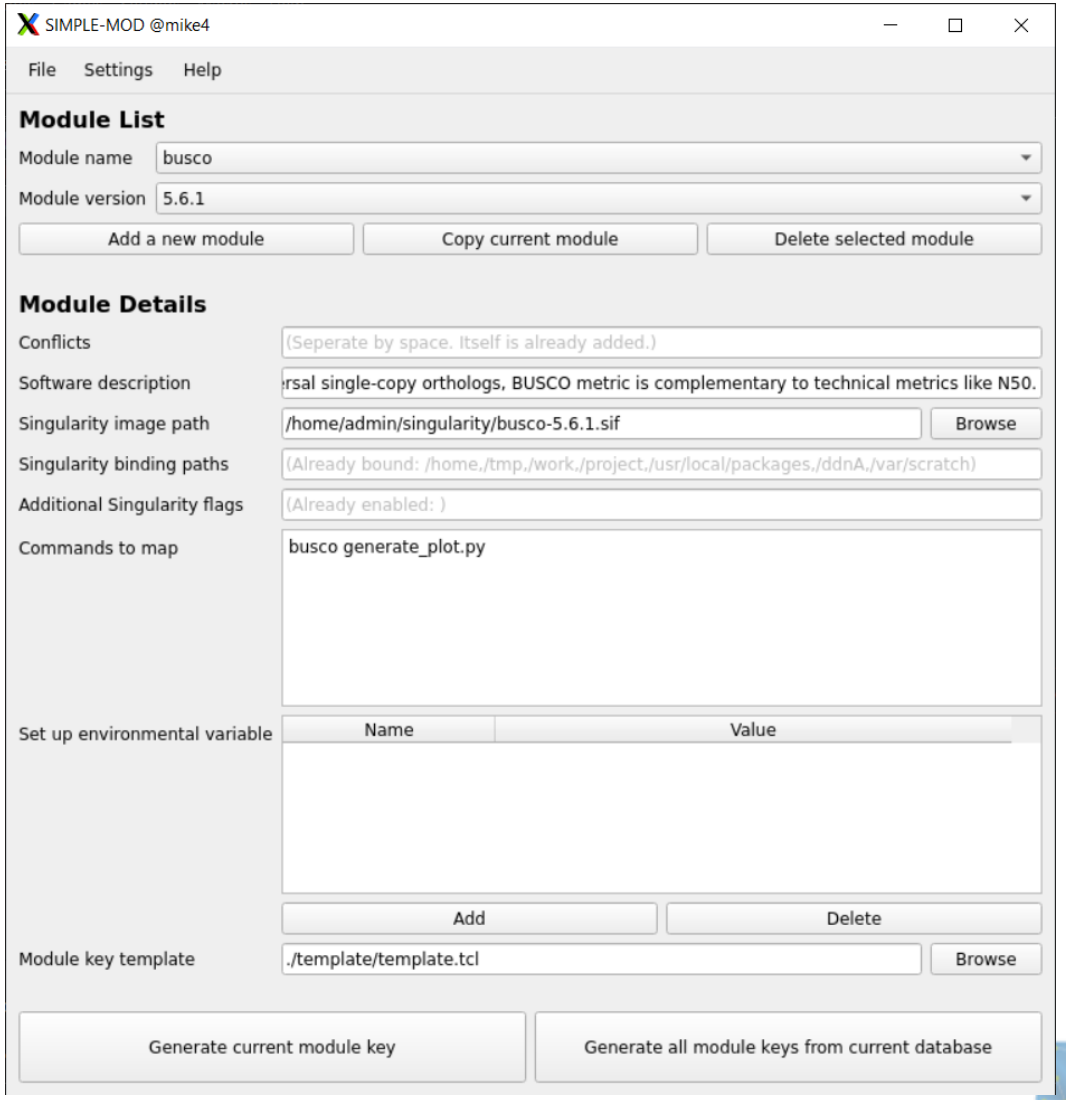
- **Contact user services**
  - Email Help Ticket: sys-help@loni.org
  - Telephone Help Desk: +1 (225) 578-0900

“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
```