HPC User Services Training

Magic Tools to Install & Manage Software · Part 1

Singularity Containers on HPC

```
$ singularity exec my_software.sif ./run.sh
```



Siva Prasad Kasetti

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

Louisiana State University, Baton Rouge · 21 October 2026

Magic Tools to Install & Manage Software

Two sessions, one goal: get the software you need running on our clusters — without admin rights.



Singularity Containers

TODAY



Conda Virtual Environments

Companion session

Next week

Today's Roadmap

01

Why Containers?

- The problems
- Containers & Singularity

02

Run an Existing Image

- What you need
- Basic commands
- Running jobs

03

Get More Images

- What you need
- Where to get
- How to get

04

Build Your Own Image

- What you need
- Typical workflow
- Recipes

Short break after Part 2 · Live demos in every part · Wrap-up & Q&A at the end

01

Why Containers?

- The problems with installing software on HPC
- What a container is — and what Singularity is

The Core Problem

You have an account, an allocation, and you can run jobs.

But the software you need isn't installed.

Installing software on a shared HPC system is harder than on your own laptop.

compile from source

```
# The traditional Linux way:
$ ./configure --prefix=$HOME/sw
$ make -j 8
$ make install

configure: error: libfoo >= 2.3
  not found
```

Compiling from source works... until it doesn't.

Four Pain Points

1 Dependencies

Most software needs other libraries and tools. If they're missing, it won't install or run.

2 Version Conflicts

Different packages need different versions of the same dependency in one shared environment.

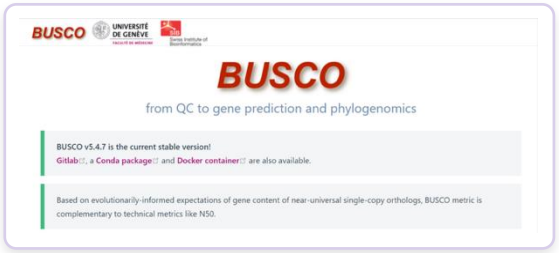
3 Root Permission

Installing often needs admin (sudo) access — which HPC users don't have.

4 Reproducibility

Re-creating or sharing the same setup with others is surprisingly hard.

1 · Dependencies

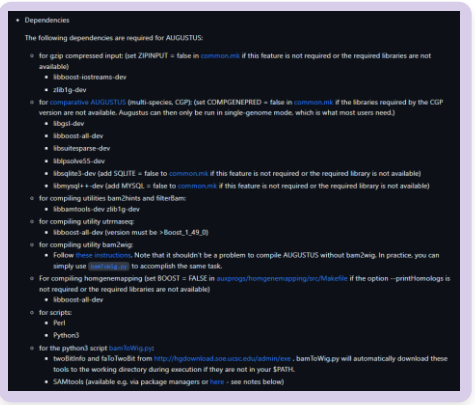


Windows: run the .exe, click Next, Next, Finish.

Linux: install every dependency first — and their dependencies too.



BUSCO's third-party dependencies



...one of them, Augustus, has its own list

Welcome to Linux!



Dependencies have dependencies. Compiling them all by hand is a rabbit hole.

2 · Version Conflicts

ML and scientific apps pin specific versions of Python, CUDA, TensorFlow, PyTorch…

Tool A

needs TensorFlow 2.10

→ `Python 3.7 – 3.10`



Tool B

needs newer package

→ `Python ≥ 3.11`

One shared environment → which Python do you install?

3 · Permission Denied

SuperMike 3

```
[kasetti@mike003 ~]$ module load python
[kasetti@mike003 ~]$ 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'
```

The Python package needs a system library that isn't installed.

`missing: libgdal`

Welcome to HPC!

3 · Permission Denied — The Usual Advice

Ask Google or ChatGPT and you'll get:

```
$ sudo yum install libgdal-devel    # RHEL
$ sudo apt-get install libgdal-dev  # Ubuntu
```

Correct — if you own the machine.

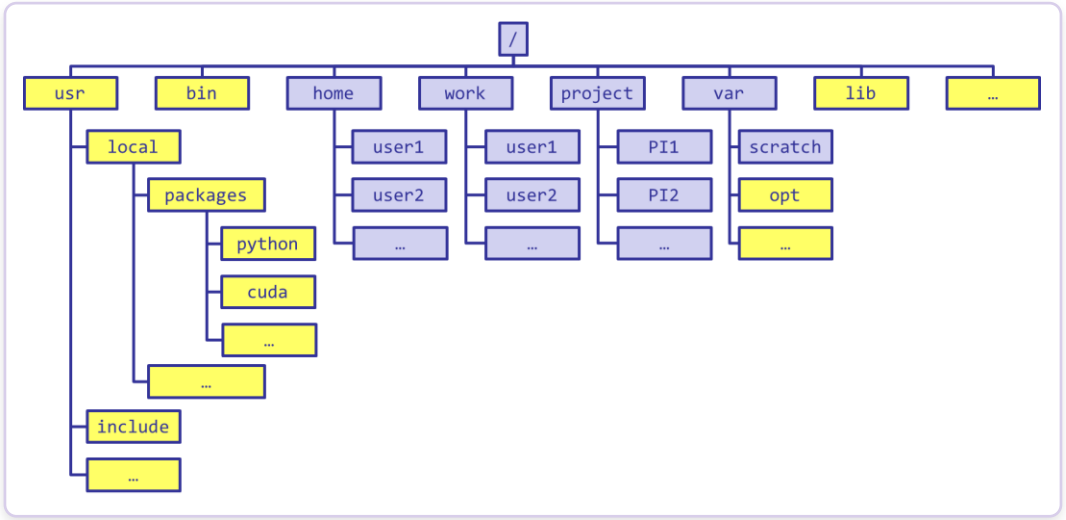
It assumes superuser rights and write access to system paths like /usr and /lib.



On HPC you don't own the computer.

You're one of many users sharing it — no sudo, no yum, no apt.

Where Can You Actually Write?



Permission	Your dirs	Others' dirs	System dirs
Read	✓	✗	✓ (some)
Write	✓	✗	✗
Execute	✓	✗	✓ (some)

Your dirs: /home, /work, /project

System dirs: /usr, /bin, /lib, …

Most directories are not writable — so system-wide installs are off the table.

4 · Reproducibility — Sharing & Migrating

Say you finally got it installed. Are you done forever?



Huge effort

Hours of work and lots of disk quota to install once.



Your colleagues?

Does everyone in your group repeat the painful install?



Another cluster?

Moving to a new system or institution means starting over.

Do Any of These Apply to You?

- [x] Dependencies
- [x] Version Conflicts
- [x] Root Permission
- [x] Reproducibility

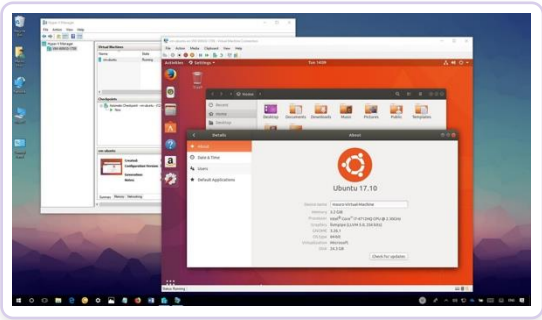
Magic tools to install & manage software

CONDA Virtual Environment

Singularity Container

Today's focus ↑

What is a Container?



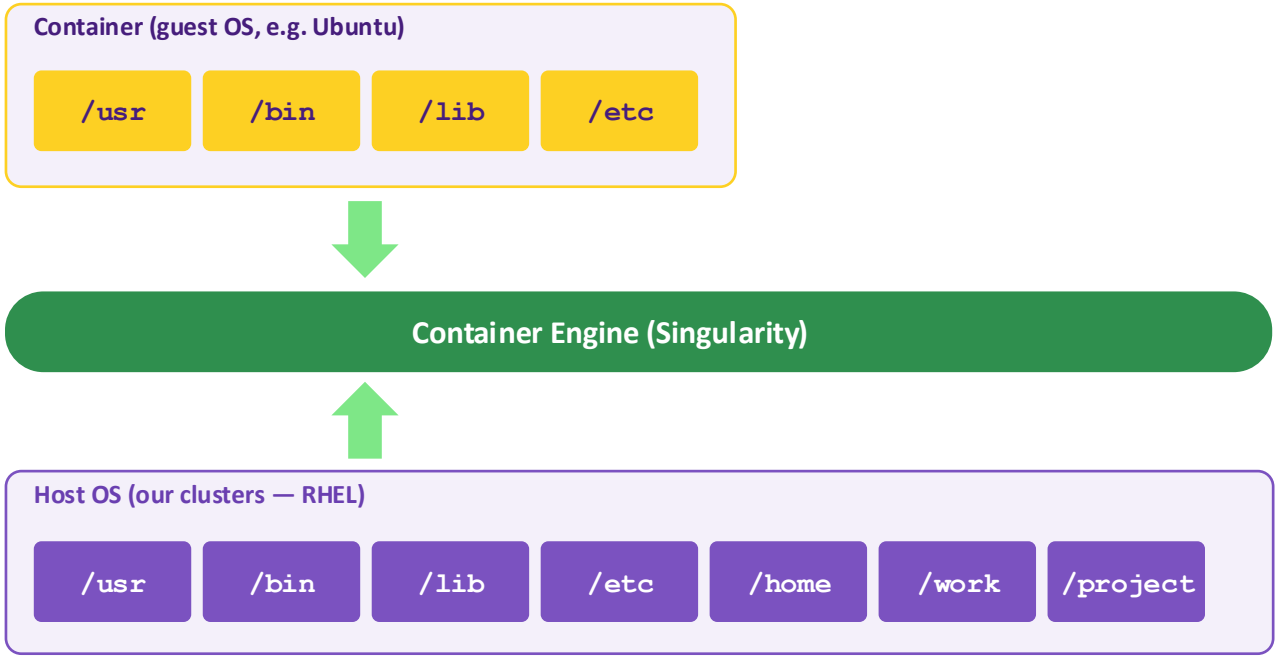
A virtual machine "mimics" an entire computer — hardware and software — inside another computer.

	Virtual Machine	Container
Virtualizes	Hardware + full OS	OS only (shares host kernel)
Size	Gigabytes	Megabytes – few GB
Startup	Minutes	Seconds
Scope	Any OS on any OS	Linux on Linux
Best for	Multiple OSes, strong isolation	App deployment, cloud, HPC

Container is
1. **a lightweight**
2. **fast "virtual machine"**

Which is fast enough for high-performance computing.

How a Container Works?



Container

A collection of files and their hierarchy — the system files of a "guest" OS plus your software.

Engine

At run time, swaps the container's files in for the host's — while your /home, /work, /project stay visible.

The “Chimera” System — in One File

What you see when the container runs:

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



- Can virtualize an entirely different OS
- Can carry software, dependencies and environment settings installed in that guest OS

`/usr` `/bin` `/lib` `/etc`

Container image

`*.sif  (older: *.simg)`

The whole OS + software stack in a single file

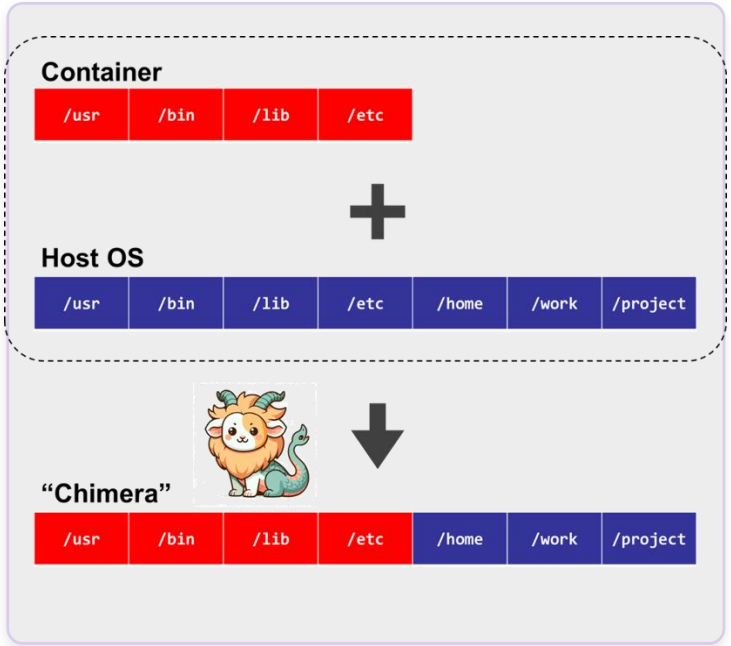
Two Key Properties

Self-contained

You virtualize an entire OS, so every dependency can live inside the container.

Isolated

Whatever happens in a container stays in that container. Break it — your host is fine. Run as many as you like side by side.



How Containers Solve Our Problems?

Dependencies	Pack every dependency — even the OS — in the container. Use apt-get / yum inside. Many developers now ship ready-made containers!
Permissions	You can't write to /usr on HPC — but you CAN inside your container.
Conflicts	Put conflicting tools in different containers.
Share / migrate	Copy the image file. Done.

What Is Singularity?

Container

A technology — an idea and an architecture.



Singularity
A software system that implements the technology.

Other implementations — like OSes: Windows, Linux, macOS…

docker

podman

kubernetes

containerd

Singularity vs Docker



- ✗ Needs root privileges
- ✗ Great on your laptop
- ✗ Not suitable for shared HPC systems



- ✓ Does NOT need root to run
- ✓ “The container for HPC”
- ✓ Runs Docker images (converts them)
- ✓ Offered on most HPC centers — including ours

Part 1 Recap

Container

A technology that helps you install and manage software — self-contained and isolated.

Singularity

The software system that implements it on our clusters — no root needed to run.

Heads-up: in 2021 Singularity split into two projects — Apptainer and SingularityCE. Same usage, compatible files. Today we just say “Singularity.”

02

Run an Existing Container Image

- What you need
- Basic commands: shell · exec · run
- Running interactive & batch jobs

Singularity on Our Clusters

On all clusters

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

Compute nodes only

Not on head/login nodes — start an interactive or batch job first.

For all users

Active account + active allocation. No extra action needed.

```
$ singularity --version          # on a compute node
$ singularity help <command>       # details for any command
```

Ready-to-Use Images

```
$ ls /project/containers/images      # on every cluster
```

```
[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
```

Today's test image

`/project/containers/images/singularity-training-ubuntu.sif`

Contents differ per cluster.

Usage 1 · Open a Shell in the Image

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

Starts an interactive shell inside the image — you're "in" the guest OS.

Option	What it does	Example
`-B <path>`	Bind host path(s) into the container. /home is bound by default; add /work and /project yourself.	`-B /work,/project`
`--nv`	Enable NVIDIA GPUs (use on a GPU node).	`--nv`

try it

```
$ singularity shell -B /work,/project $IMG
Singularity> cat /etc/os-release      # Ubuntu, not RHEL!
```

Usage 2 & 3 · exec and run

exec

```
singularity exec [options]
  <container> <command>
```

Execute a single command in the image.

Most common — works in interactive and batch jobs.

```
$ singularity exec $IMG python3
myscript.py
```

Same options as shell: -B, --nv

run

```
singularity run [options]
  <container>
```

Run the container's prewritten script.

Less common — only if the author wrote one.

```
$ singularity run $IMG
```

Demo · Run an Existing Image

LIVE DEMO

LSU

demo-1: run

```
# 1. head node — no singularity here
$ singularity --version
# 2. get a compute node
$ salloc -A <alloc> -p single -N1 -n4 -t 1:00:00
$ IMG=/project/containers/images/singularity-training-
ubuntu.sif
# 3. shell in, compare OS & tools
$ singularity shell $IMG
# 4. bind, exec, GPU
$ singularity shell -B /work,/project $IMG
$ singularity exec $IMG python3 --version
$ singularity exec --nv $IMG nvidia-smi
```

Watch for

- Singularity only exists on compute nodes
- /etc/os-release: Ubuntu inside, RHEL outside
- apt exists inside; gcc from the host doesn't
- /work is invisible until you bind it
- exec: everything after the image is your normal command
- --nv exposes the GPU

LSU INFORMATION TECHNOLOGY SERVICES

LONI

Running Jobs with Singularity

Job type	Commands	Purpose
Interactive	`singularity shell …` `singularity exec …`	Debugging & testing
Batch	`singularity exec …`	Production runs



No shell in batch jobs — there's no one to type into it. Use exec.

Batch Job · Before and After

regular.sh

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

cd /path/to/work/dir

python myjob.py
```

container.sh

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

cd /path/to/work/dir

IMG=/project/containers/images/
    singularity-training-ubuntu.sif
singularity exec -B /work,/project \
    $IMG python myjob.py
```

Only the last lines change — the python now comes from the image.

Part 2 Recap · Three Commands

shell	`singularity shell [options] <container>`	Start an interactive shell in the image
exec	`singularity exec [options] <container> <command>`	Execute a command in the image
run	`singularity run [options] <container>`	Run the image's prewritten script

Anyone with an active account and allocation can run images already on the cluster — no extra permission.



5-Minute Break

```
$ sleep 300   # then Part 3 — Get More Images
```

Questions so far? Grab me during the break.

03

Get More Container Images

- What you need
- Where to get images you can trust
- How to get them: pull & build

What You Need to Bring Your Own Image

BEFORE

- ~~Be added to the "singularity" group~~
- ~~Email sys-help@loni.org to request~~
- ~~chgrp singularity <image> every time~~

NOW

- Active account + allocation
- No group membership needed
- No chgrp step

Pull or upload an image to your space and run it — the same as a pre-installed one.

Where to Get Images — Carefully

You can get images from many places. Not that you should!

Reliability Third-party or untested images may simply not work.

Security risk Untrustworthy publishers can pack in things you don't know about.

Solution Always get images from a source you can trust.



techradar.com — malware attacks on Docker Hub spread millions of malicious repositories

Trusted Sources, in Order

TIER 1

Developer's official release

On the software's website or GitHub, look for “Docker”, “Singularity” or “Container”.

E.g. TensorFlow, Trinity, Salmon

TIER 2 · Trustworthy third parties

Source	Good for
BioContainers	Bioinformatics tools · biocontainers.pro
NVIDIA NGC	GPU-optimized AI/HPC · catalog.ngc.nvidia.com
Bitnami	Common app stacks (by VMware)
Docker Hub / Quay.io	Huge — but anyone can publish. Prefer Official / Verified publishers; avoid unknown uploaders.

Look for: Docker Official Image Verified Publisher

How to Get · singularity pull

```
$ singularity pull [options] [target.sif] <source>
```

<source>	What it does
`docker://name[:tag]`	Pull a Docker image and convert it to .sif. Many official releases are Docker images. (Like: docker pull name[:tag])
`https://site/image.sif`	Download an image file from a web source



Tip: pulls cache layers in ~/.singularity and can fill your home quota.

```
export SINGULARITY_CACHEDIR=/work/$USER/.sing_cache
```

How to Get · singularity build (advanced)

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



docker://name[:tag]
Docker image (same as pull)



image.sif
Another local image file



sandbox_dir/
A sandbox — a container as a directory



recipe.def
A recipe — instructions to build an image

Target filename is required. Sandbox & recipe come back in Part 4.

For pulling from Docker Hub, pull and build are interchangeable.

Demo · Pull from Docker Hub

LIVE DEMO

LSU

demo-3: pull

```
# on a compute node, in /work
$ export SINGULARITY_CACHEDIR=/work/$USER/.sing_cache
$ singularity pull docker://ubuntu
$ singularity pull docker://ubuntu:20.04
$ singularity pull myubuntu.sif docker://ubuntu:20.04
# a real tool: Salmon (COMBINE-lab)
$ singularity pull docker://combinelab/salmon
$ singularity exec salmon_latest.sif salmon --version
# same thing with build
$ singularity build u2004.sif docker://ubuntu:20.04
```

Watch for

- Default tag is :latest
- Output name = <name>_<tag>.sif
- Give your own filename as target
- Read the tool's docs — container releases often live on Docker Hub
- The pulled image runs right away

LSU INFORMATION TECHNOLOGY SERVICES

LONI

Bonus · Hot Packages with GPU Support

PyTorch

```
$ singularity pull docker://pytorch/pytorch:2.14.1-cuda12.6-cudnn9-devel
```

Easier: `$ module load pytorch` — our container-based module

TensorFlow

```
$ singularity pull docker://tensorflow/tensorflow:2.22.0rc0-gpu-jupyter
```

Easier: `$ module load tensorflow` — our container-based module

Tags change fast — check Docker Hub for the latest before you pull.

04

Build Your Own Container Image

- What you need
- Typical workflow: sandbox → image
- Make it easier: recipes

When to Build — and the Big Idea

1. No container release exists — need to DIY
2. Install would be easy with sudo apt / sudo yum — if I had permission
3. Found a container, but need to add something (e.g., a Python module)



Your local Linux machine

`sudo · apt-get · yum` ✓

build & transfer the .sif



HPC systems

run it — as if you had installed with sudo

What You Need — the Full Picture

Run an existing image	Account + active allocation
Get more images	Same as above (no group needed now)
Build your own image	**+ a local Linux machine where you have root**

No Linux box? Use WSL2 on Windows or a Linux VM on macOS.

Install Singularity on Your Machine



compatible



- ✓ Joined the Linux Foundation
- ✓ Easier install (e.g., apt on Ubuntu)

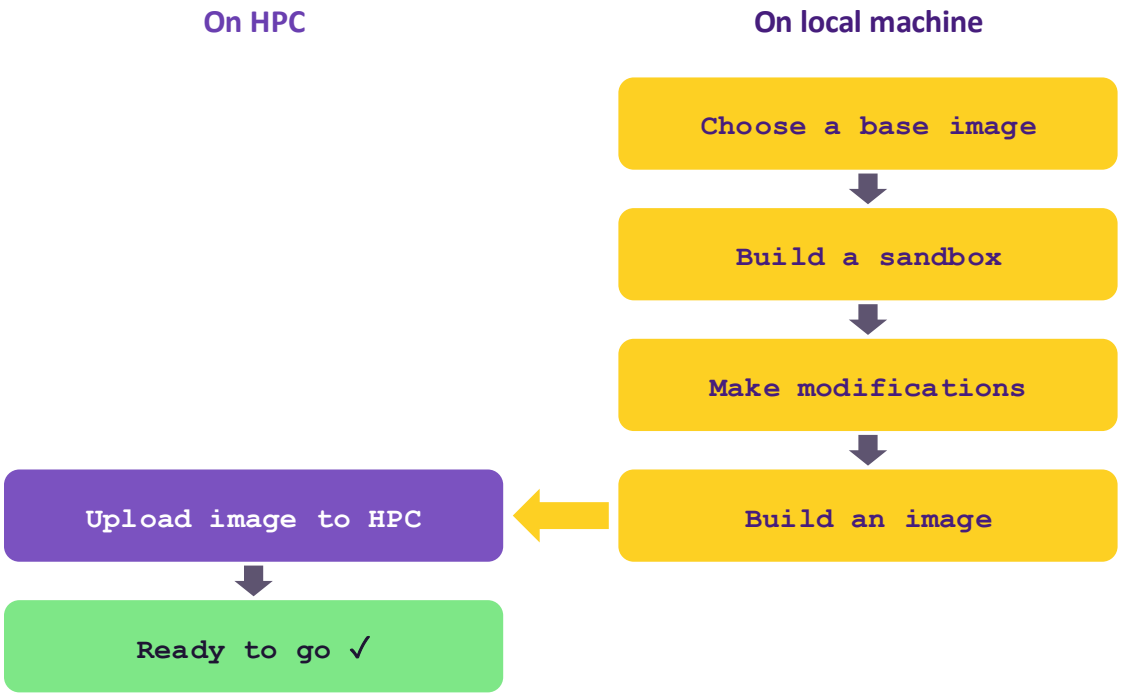
apptainer.org/docs/admin/main/installation.html

- ✓ Sylabs — community supported
- ✓ What's installed on our clusters

docs.sylabs.io/guides/latest/admin-guide/

Either works — same commands, same .sif files. Apptainer is usually the easier install.

Typical Workflow



Remember

Build locally (where you have root). Upload. Run on HPC.

Testing is implied at every step.

Step 1 · Choose a Base Image



A minimal, “mint” OS

`Ubuntu, Rocky, Debian, …`

No image has your software yet — install from scratch. Keeps the image small.



An image with software installed

`TensorFlow, PyTorch, …`

Modify a working image — e.g., add a Python module to TensorFlow.

Step 2 · Build a Sandbox

What's a sandbox?

A container in directory form (not a single file) — so you can modify it.

local machine

```
$ singularity build --sandbox
    ubuntu/ docker://ubuntu:24.04

$ ls ubuntu/
bin  boot  dev  etc  home  lib  ...
```

Source can be any build source: docker://, a .sif, another sandbox, or a recipe.

Step 3 · Make Modifications

```
$ sudo singularity shell --writable ubuntu/
```



sudo

Root inside the container
Needed for apt / yum and system files — which is why this happens on your own machine.



--writable

Write to the sandbox
Without it the sandbox is read-only.

inside the sandbox

```
Singularity> apt update
Singularity> apt install -y bat          # anything you like
```

Step 4 · Build an Image from the Sandbox

```
$ sudo singularity build bat.sif ubuntu/
```

Modified with sudo? Build with sudo!

To …	Run …
Build a sandbox	`singularity build --sandbox …`
Modify the sandbox	`sudo singularity shell --writable …`
Build image from sandbox	`sudo singularity build …`

Demo · Sandbox to Image to HPC

LIVE DEMO

LSU

demo-4: build

```
# on the local Linux machine
$ singularity build --sandbox ubuntu/ docker://ubuntu:24.04
$ singularity shell --writable ubuntu/  # apt fails
$ sudo singularity shell ubuntu/        # read-only
$ sudo singularity shell --writable ubuntu/
Singularity> apt update && apt install -y bat
Singularity> batcat <file_name>
$ singularity build bat.sif ubuntu/     # fails
$ sudo singularity build bat.sif ubuntu/
$ scp bat.sif <user>@<cluster>:/work/<user>/
# on HPC compute node
$ singularity exec bat.sif batcat --help
```

Watch for

- --writable alone: can write, no permission
- sudo alone: permission, but read-only
- Both together: apt works
- Build without sudo fails after a sudo modify
- Same image runs on HPC — no sudo there

LSU INFORMATION TECHNOLOGY SERVICES

LONI

Make It Easier · Why a Recipe?

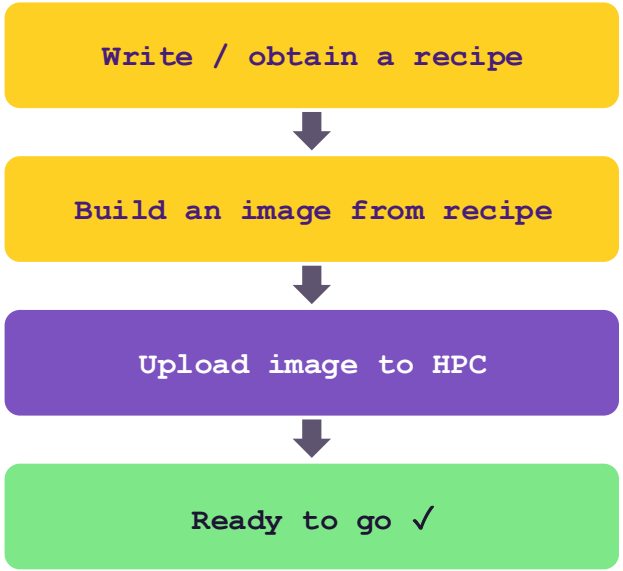
The sandbox workflow

Pro
Flexible — try things interactively

Cons
Hard to repeat; trial-and-error bloats image size

Recipe: a text file with the instructions to build a container. Repeatable, shareable, lean.

The recipe workflow



Anatomy of a Recipe · recipe.def

recipe.def

```
Bootstrap: docker
From: ubuntu:24.04

%labels
    Author Siva Prasad Kasetti
    Version 1.0
    Description Install batcat

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

%environment
    export MYENV="My environment variable"

%runscript
    echo "Print bat version:"
    batcat --version
```

`Header`	Base image: how, where, what to pull
`%labels`	Container info — write whatever you want
`%post`	Commands run after the base image is pulled
`%environment`	Variables set every time the container runs
`%runscript`	What singularity run executes

Build from the Recipe

```
$ singularity build bat.sif recipe.def
```



No sudo needed

Apptainer builds from a recipe in “fakeroot” mode. If your install complains, add --fakeroot or use sudo.



Still build locally

Fakeroot builds aren't enabled on our clusters — build on your machine, then upload the .sif.

Demo · Recipe to HPC

LIVE DEMO

LSU

demo-5: recipe

```
# local machine
$ cat recipe.def
$ singularity build bat.sif recipe.def
$ scp bat.sif <user>@<cluster>:/work/<user>/

# HPC compute node
$ singularity shell bat.sif
Singularity> echo $MYENV
$ singularity exec curl.sif batcat --version
$ singularity run bat.sif
$ singularity inspect bat.sif  # labels
```

Watch for

- One command builds the whole image
- %environment → $MYENV is set
- %runscript → what run prints
- %labels → shown by inspect
- Recipe = a record you can re-run and share

LSU INFORMATION TECHNOLOGY SERVICES

LONI

Take-Home Messages



Run

Anyone with an account & allocation can run images already on the cluster.

`shell · exec · run`



Get more

Pull from trusted sources — developers first, reputable third parties next.

`pull · build`



Build your own

Need a Linux machine with root. Build locally, upload, run.

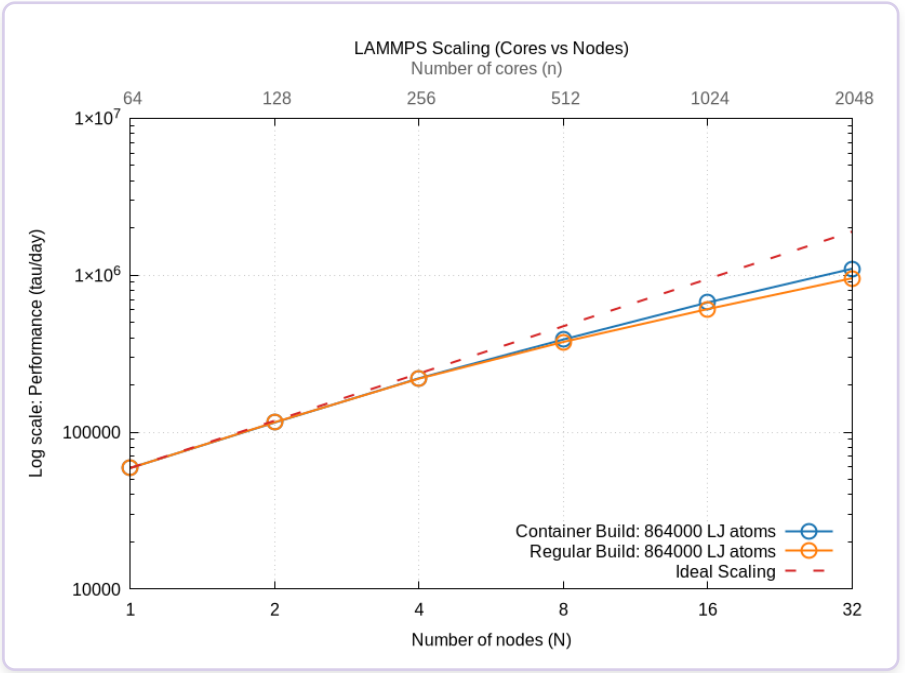
`sandbox · recipe`

Build on your machine → upload to HPC → run.

Does a Container Slow You Down?

No.

- Benchmark: LAMMPS molecular dynamics, scaled across nodes
- Compared a regular build vs the same code in a container
- **Performance was identical**



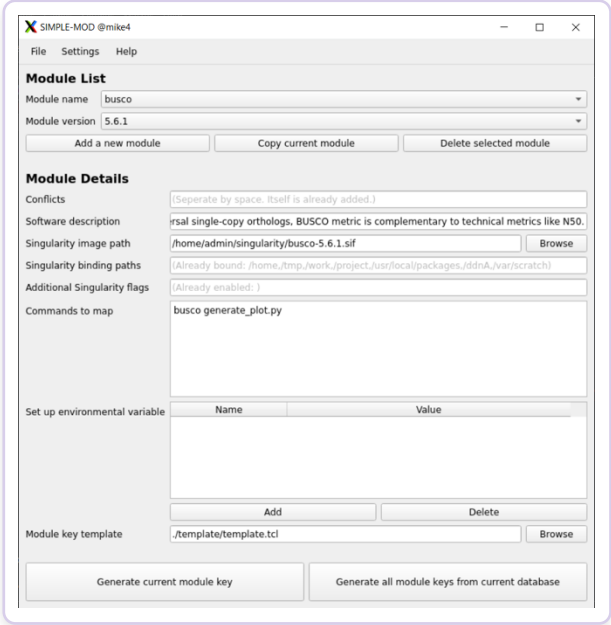
“Commercial” Time · SIMPLE-MOD

Tired of typing this?

```
$ singularity exec --nv -B /work,/project,\
  /usr/local/packages /path/to/busco.sif \
  busco --version
```

Load it like any module:

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



github.com/lsuhpchelp/SIMPLE-MOD

We Have Covered Today

01

Why Containers?

- The problems
- Containers & Singularity

02

Run an Existing Image

- What you need
- Basic commands
- Running jobs

03

Get More Images

- What you need
- Where to get
- How to get

04

Build Your Own Image

- What you need
- Typical workflow
- Recipes

```
$ singularity exec questions.sif ask --anything
```

Questions?



Help ticket
sys-help@loni.org



Help desk
+1 (225) 578-0900

Recordings and slides are posted on the LSU HPC training website. Thank you!