

# Introduction to Containers for Scientific Container-Native Workflows: **Singularity** on **ACES**

Richard Lawrence  
03/24/2026



High Performance  
Research Computing  
DIVISION OF RESEARCH



*developed for*



# Outline

- Overview of Containers
- Overview of Singularity
- Getting Started
- Container Image Sources
- Working with Images
- Working with Containers
- Containerized Scientific Applications on ACES
  - PyTorch
  - LAMMPS

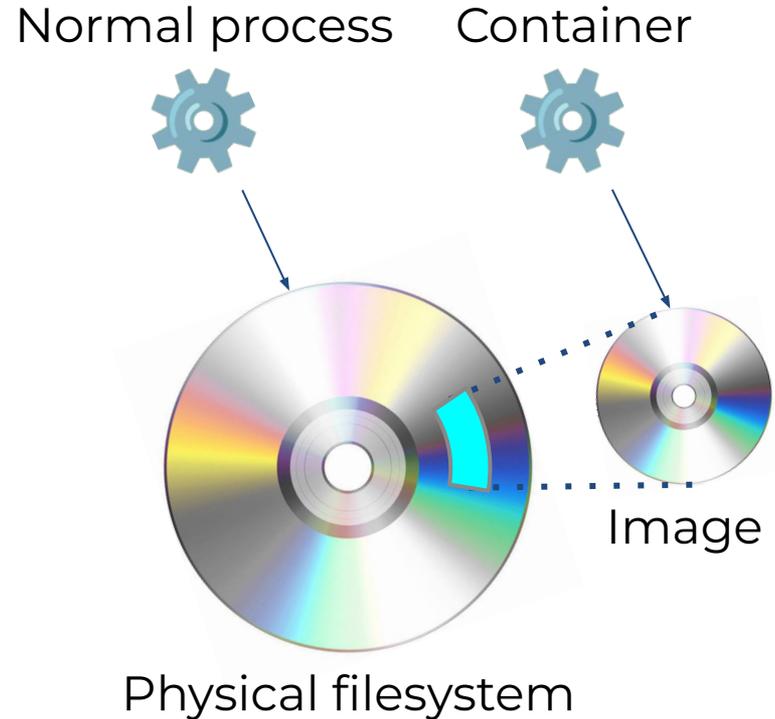
# Learning Resources

- Slides on the course web page  
[https://hprc.tamu.edu/training/aces\\_singularity.html](https://hprc.tamu.edu/training/aces_singularity.html)  
(highly recommended for working along)
- HPRC's Knowledge Base  
<https://hprc.tamu.edu/kb/Software/Singularity/>
- HPRC on YouTube  
<https://www.youtube.com/@TexasAMHPRC/playlists>
- ACCESS Links  
<https://support.access-ci.org/ci-links>

# Overview of Containers

# What Are Containers?

- A container is a process (⚙️) that has its own **view** of local resources:
  - **Filesystem**
  - User IDs
  - Network
  - etc.
- Example: this container (⚙️ on the right) sees the **image** instead of the physical filesystem



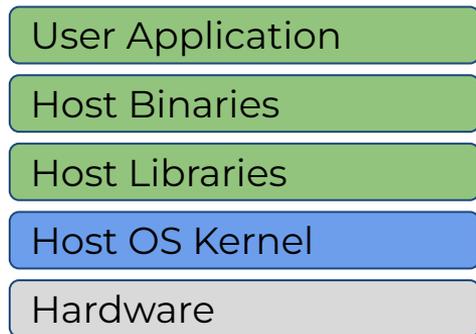
# Why Use Containers?

- **Shareability:**
  - Share your container image file by uploading to a public repository
  - Use images shared by others
- **Portability:**
  - Use images on any computer with the same architecture (x84-64)
- **Reproducibility:**
  - Container users are largely unaffected by changes to the cluster environments

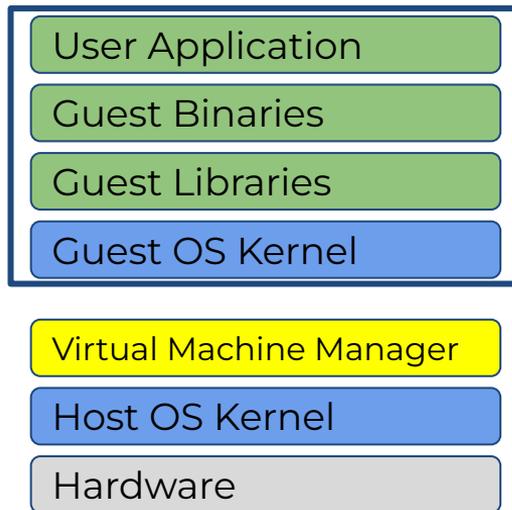
# What Goes In Container Images?

- Unlike in VMs, the OS Kernel is not duplicated
- Container images are smaller than VM images

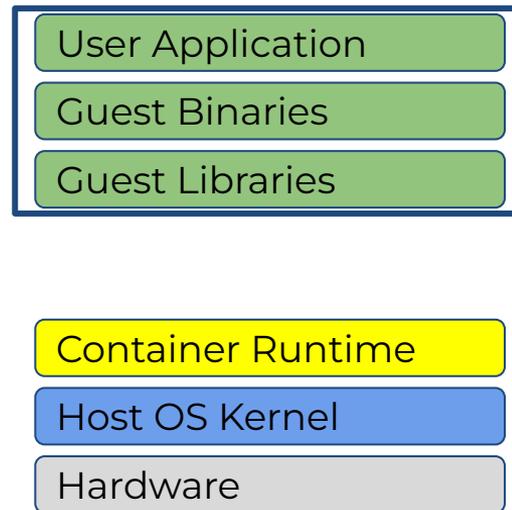
## Local Build, or “Bare metal”



## Virtual Machine

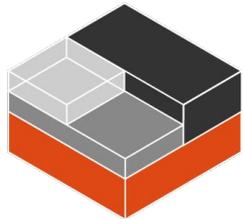


## Container



# Popular Container Runtimes

Instant deployment to users on different devices!



LXC  
2008



docker

Docker  
2013

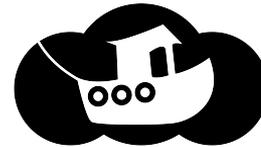


Singularity  
2015



SHIFTER

Shifter 2016



Charliecloud

Charliecloud  
2017



Podman  
2018

# Overview of Singularity

# Singularity

- An easy-to-use, high-performance container solution



**Deploying Secure Container  
Solutions from Edge to Exascale**

Presented by



# Singularity is Apptainer



# Singularity Features

- Singularity is a container runtime and an image builder
- Singularity can read and convert Docker images
- Filesystem inside container is isolated
- User inside container is the same as the user outside
- Works with high-performance cluster technologies

Read more in the Apptainer manual

<https://apptainer.org/user-docs/3.8/>

# Singularity on ACES

- Singularity is available on Compute nodes
  - Singularity activities are too cpu-intensive for login nodes.
- Singularity images can be large on disk. Be aware of your storage quota. (`/scratch` > `/home`)
- Some container activities may be too I/O-intense for the shared network filesystem. Be courteous to others and use a local filesystem for large image operations.

# Getting Started

# ACES Portal

TEXAS A&M HIGH PERFORMANCE RESEARCH COMPUTING

Home User Services Resources Research Policies Events Training About **Portal**

Quick Links

- New User Information
- Accounts
- Apply for Accounts

normalized vorticity( $\zeta/f$ )

-2 -1 0 1 2

Grace Portal

FASTER Portal

**ACES Portal (ACCESS)**

Launch Portal (ACCESS)

ACES Portal [portal-aces.hprc.tamu.edu](http://portal-aces.hprc.tamu.edu)  
is the web-based user interface for the ACES cluster

Open OnDemand (OOD) is an advanced web-based  
graphical interface framework for HPC users

ACES OnDemand Portal

ACES

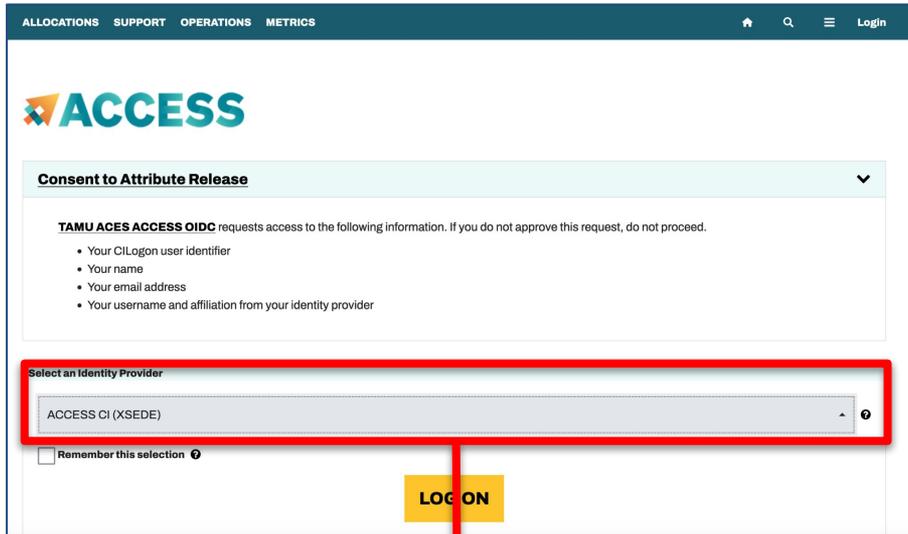
ACCELERATING COMPUTING  
FOR EMERGING SCIENCES

OnDemand provides an integrated, single access point for all of your HPC resources.

powered by  
OPEN OnDemand

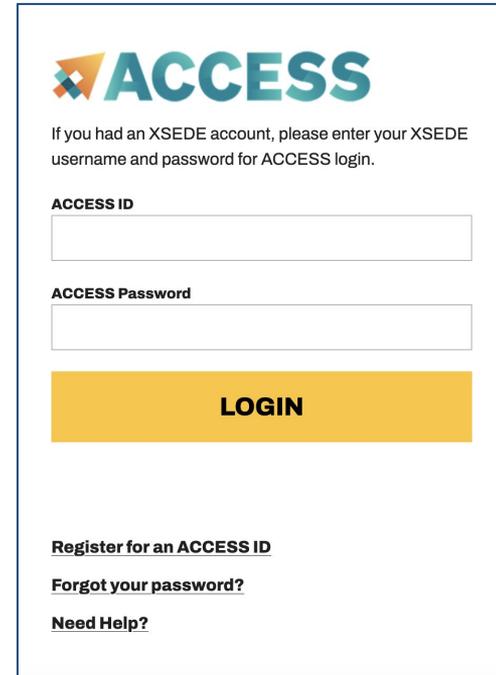
OnDemand version: 3.0.0

# Accessing ACES via the Portal (ACCESS)



The screenshot shows the ACCESS portal interface. At the top, there is a navigation bar with links for ALLOCATIONS, SUPPORT, OPERATIONS, METRICS, and a Login button. Below the navigation bar is the ACCESS logo. A section titled "Consent to Attribute Release" contains a message from TAMU ACES ACCESS OIDC and a list of requested information: CI/Logon user identifier, name, email address, and username/affiliation. Below this is a "Select an Identity Provider" dropdown menu with "ACCESS CI (XSEDE)" selected. A red box highlights this dropdown and the "LOG ON" button below it. A red line connects the "LOG ON" button to the text below.

Select the Identity Provider appropriate for your account.

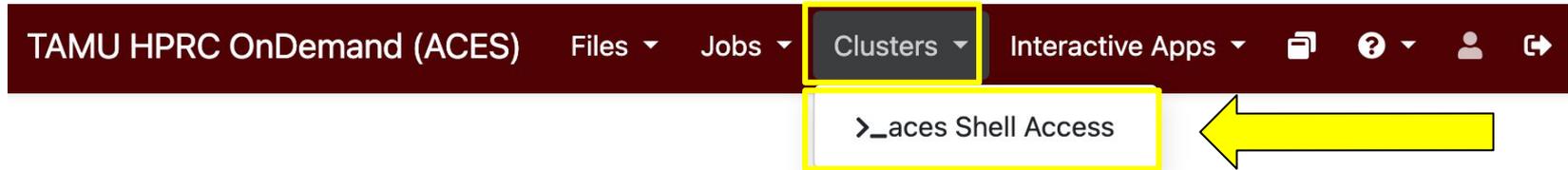


The screenshot shows the ACCESS portal login form. It features the ACCESS logo at the top, followed by a message: "If you had an XSEDE account, please enter your XSEDE username and password for ACCESS login." Below this are two input fields: "ACCESS ID" and "ACCESS Password". A large yellow "LOGIN" button is positioned below the password field. At the bottom, there are links for "Register for an ACCESS ID", "Forgot your password?", and "Need Help?".

Log-in using your ACCESS or institutional credentials.

# Get a Shell on ACES

Click on “Clusters” menu →>\_aces Shell Access



# Success!

Welcome to the  
ACES login node.

```
portal-aces.hprc.tamu.edu/pun/sys/shell/ssh/login.aces
Host: login.aces Themes: Default

=====
|               Texas A&M University High Performance Research Computing               |
|-----|
| Website:                https://hprc.tamu.edu                                     |
| Consulting:             help@hprc.tamu.edu (preferred) or (979) 845-0219          |
| ACES Documentation:    https://hprc.tamu.edu/kb/User-Guides/ACES                 |
| FASTER Documentation:  https://hprc.tamu.edu/kb/User-Guides/FASTER              |
| Grace Documentation:   https://hprc.tamu.edu/kb/User-Guides/Grace               |
| Terra Documentation:   https://hprc.tamu.edu/kb/User-Guides/Terra                |
| YouTube Channel:       https://www.youtube.com/texasamhprc                     |
|-----|
|=====|
|*****|
|      *      == IMPORTANT POLICY INFORMATION ==      *      |
|      *      - Unauthorized use of HPRC resources is prohibited and subject to    *      |
|      *      - Use of HPRC resources in violation of United States export control *      |
|      *      - Sharing HPRC account and password information is in violation of  *      |
|      *      - Authorized users must also adhere to ALL policies at:            *      |
|      *      https://hprc.tamu.edu/policies/                                       *      |
|      *      *****                                                                *      |
|      *      !! WARNING: THERE ARE ONLY NIGHTLY BACKUPS OF USER HOME DIRECTORIES. !! *      |
|      *      Please restrict usage to 8 CORES across ALL login nodes.             *      |
|      *      Users found in violation of this policy will be SUSPENDED.          *      |
|      *      To see these messages again, run the motd command.                  *      |
|      *      [u.rl117197@aces-login2 ~]$ █                                         *      |
|*****|
```

# Set Up Your Tutorial Environment

```
cd $SCRATCH  
mkdir s_tutorial  
cd s_tutorial  
pwd
```

```
export TRAINING=/scratch/training/singularity  
ls $TRAINING
```

# Set Up Your Singularity Environment

Get to a compute node from the login node

```
srun --time=120 --ntasks=1 --mem=4G --pty bash -i
```

Return to your tutorial directory (if necessary)

```
cd $SCRATCH/s_tutorial
```

following along live? add:  
--reservation=containers

Set your singularity cache directory for temporary files

```
export SINGULARITY_CACHEDIR=$TMPDIR
```

Connect to the internet for fetching images

```
module load WebProxy
```

# Your First Singularity Container

Singularity can fetch an image *and* launch a shell in one line.

```
singularity shell --help
```

Fetch an image and launch a shell from it

```
singularity shell docker://almalinux:8  
cat /etc/redhat-release  
exit
```

The ACES compute nodes also have Red Hat linux installed.

```
cat /etc/redhat-release
```



**Congratulations!**

**Welcome to containers**

[WWW.FUNIMADA.COM](http://WWW.FUNIMADA.COM)

# Container Image Sources

# Popular Repositories

The most common repository is:

- Docker Hub

Others repositories include:

- Singularity Hub
- Singularity Library
- NVIDIA GPU Cloud
- Quay.io
- BioContainers

See

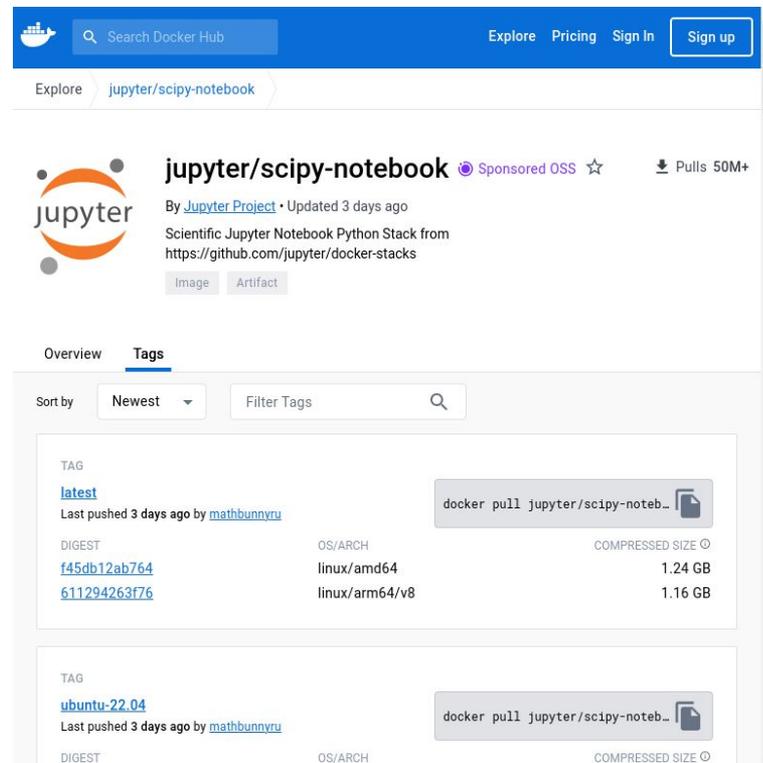
<https://hprc.tamu.edu/kb/Software/Singularity/Examples/#popular-repositories>

# Docker Hub Example

Docker Hub repositories are named in the form `<group>/<name>` similar to GitHub.

Each image within a repository has a `<tag>` that describes how and when it was built.

This example is `jupyter/scipy-notebook:latest`



The screenshot shows the Docker Hub interface for the `jupyter/scipy-notebook` repository. The page includes a search bar, navigation links (Explore, Pricing, Sign In, Sign up), and the repository details. The repository is sponsored by the Jupyter Project and has over 50 million pulls. The 'Tags' section is active, showing a list of tags sorted by 'Newest'. The 'latest' tag is highlighted, with a 'docker pull' button next to it. Below the 'latest' tag, there are two entries for different OS/architectures: 'linux/amd64' and 'linux/arm64/v8', each with a compressed size of 1.24 GB and 1.16 GB respectively. The 'ubuntu-22.04' tag is also visible below.

TAG	DIGEST	OS/ARCH	COMPRESSED SIZE
<a href="#">latest</a>	<a href="#">f45db12ab764</a>	linux/amd64	1.24 GB
	<a href="#">611294263f76</a>	linux/arm64/v8	1.16 GB
<a href="#">ubuntu-22.04</a>			

# Singularity Pull

Singularity can fetch images from repositories and also convert them to the singularity file format at the same time.

```
singularity pull [target-filename] <source>
```

Where <source> refers to something on the internet. The syntax depends on where the source is located.

and [target-filename] includes the file extension.

# Singularity Pull Example

The `<source>` argument for Docker images looks like

```
docker://<group>/<name>[:<tag>]
```

Therefore the pull command for the Jupyter example is,

```
singularity pull docker://jupyter/scipy-notebook:latest
```

The default filename will be `scipy-notebook_latest.sif`

(Download now or copy the example<sup>1</sup>; we will need this later)

<sup>1</sup>`$TRAINING/scipy-notebook_2023.sif`

# Working with Images

# Singularity Image Formats

- Singularity container images come in two main formats:
  1. Directory
  2. Single file. Singularity uses the SIF format for single file images. This is the default.
- The `singularity build` tool can convert images in both formats.  
`singularity build --help`
- The `--sandbox` option is used to create directory-format images.

# Singularity Image Exercise

Singularity pull can fetch an image and write to either file format.  
*(note the order of the arguments)*

```
singularity pull almalinux.sif docker://almalinux:8
```

Singularity can convert an image to the directory file format.  
Use the `--sandbox` argument to specify the directory type.  
*(note the order of the arguments)*

```
singularity build --sandbox $TMPDIR/almalinux almalinux.sif
```

# Singularity Write Exercise

Directory images are writable. Simply add the `--writable` flag to your container command.

```
singularity shell --writable $TMPDIR/almalinux  
mkdir /my_dir  
exit
```

Are the changes still there?

```
singularity shell $TMPDIR/almalinux  
ls /
```

# Singularity Read-only Exercise

SIF files are safe for network file system /scratch.

```
singularity build --fakeroot my_almalinux.sif $TMPDIR/almalinux
```

Are the changes still there?

```
singularity shell my_almalinux.sif  
ls /  
exit
```

What about the --writable flag?

```
singularity shell --writable my_almalinux.sif  
no.
```

# Working with Containers

# Launching Processes

Singularity has three methods for launching processes:

- **Interactive**: `singularity shell`
- **Batch processing**: `singularity exec`
- **Container-as-executable**: `singularity run`

# Singularity Run Exercise

Singularity run will execute the default runscript, if one was defined. You may also execute the container directly.

```
singularity pull docker://hello-world  
singularity run hello-world_latest.sif  
Hello from Docker!  
./hello-world_latest.sif  
Hello from Docker!
```

Docker hello-world is a minimal image. This is all it can do.

# Singularity Exec Exercise

Singularity Exec lets you access executables and other commands in a container. This is appropriate for batch jobs.

ACES nodes have Python 3.

```
python3 --version  
Python 3.6.8
```

Our singularity image has a different Python 3.

```
singularity exec scipy-notebook_latest.sif python3 --version  
Python 3.11.6
```

# Working with Files

- Filesystem inside a container is isolated from the real, physical filesystem.
- To access your files, ensure the directory is *mounted*.
- By default, Singularity will mount `$HOME` and `$PWD` if it can.
- To specify additional directories, use the `SINGULARITY_BINDPATH` environment variable or the `--bind` command line option.

# Working with Files Exercise

Recommended that you mount `/scratch` to get access to your data storage, and `/tmp` to get access to the local disk on the node.

```
singularity shell --bind "/scratch,/tmp" <image>  
mkdir $TMPDIR/my_dir; exit  
ls $TMPDIR
```

Notice that your variables like `$TMPDIR` get passed into the container by default.

*(singularity on ACES already binds these directories by default)*

# Singularity Batch Example

```
#!/bin/bash

## JOB SPECIFICATIONS
#SBATCH --job-name=sing_test           #Set the job name to "sing_test"
#SBATCH --time=00:10:00                #Set the wall clock limit to 10min
#SBATCH --ntasks=4                    #Request 4 task
#SBATCH --mem=2560M                   #Request 2560MB (2.5GB) per node
#SBATCH --output=sing_test.%j         #Send stdout/err to "sing_test.[jobID]"
```

```
export SINGULARITY_BINDPATH="/scratch,/tmp"
```

```
# execute the default runscript defined in the container
singularity run hello-world_latest.sif
```

```
# execute a command within container
# specify the full path if the command is not in PATH
singularity exec scipy-notebook_latest.sif python3 hello.py
```

ONE VARIABLE

2 CONTAINERS

# Interactive Graphical Computing

The image shows a screenshot of the ACES OnDemand Portal. The top navigation bar is dark red and contains the following items: "ACES OnDemand Portal", "Files", "Jobs", "Clusters", "Interactive Apps", and "Dashboard". The "Interactive Apps" menu is open, showing a list of applications categorized into three sections: "GUI", "Imaging", and "Servers". The "Jupyter Notebook" option in the "Servers" section is highlighted with a yellow box. A red arrow points from a box labeled "click click" to the "Interactive Apps" menu, and another red arrow points from the same box to the "Jupyter Notebook" option. To the left of the menu is a large logo consisting of a central white circuit-like pattern on a dark red circle, surrounded by four overlapping colored shapes (green, blue, green, blue) in a square arrangement.

ACES OnDemand Portal Files Jobs Clusters Interactive Apps Dashboard

GUI

- VNC
- Nextsilicon VNC

Imaging

- CryoSPARC
- ImageJ
- cisTEM

Servers

- Jupyter Notebook
- JupyterLab
- RStudio

click  
click

# Containerized Jupyter Notebook

Choose *Containers*

Enter

`$SCRATCH/s_tutorial/scipy-notebook_latest.sif`  
or wherever your file actually is (see [Slide 27](#))

Backup copy at  
`/scratch/training/singularity/scipy-notebook_2023.sif`

Home / My Interactive Sessions / Jupyter Notebook

## Interactive Apps

- GUI
- VNC
- Nextsilicon VNC
- Imaging
- CryoSPARC
- ImageJ
- cisTEM
- Servers
- Jupyter Notebook

## Jupyter Notebook

This app will launch a Jupyter Notebook server on the ACES cluster.

Type of Environment

Containers (Singularity/Charliecloud)

Select the type of environment in which Jupyter is installed. [Help me choose](#)

Path to container image file

`$SCRATCH/s_tutorial/scipy-notebook_latest.sif`

Enter the full path to an image file. Recommended that this be located in your \$SCRATCH directory.

Singularity images and Charliecloud images are supported. Images should contain the Jupyter app.

# ...Continued

Launch

Jupyter Notebook (5488) 1 node | 1 core | Starting

Jupyter Notebook (5489) 1 node | 1 core | Running

Host: >\_ac110 Delete

Created at: 2023-09-21 15:39:52 CDT

Time Remaining: 56 minutes

Session ID: a5f41dfd-7c0d-44e3-aea7-7331c66a4d24

Connect to Jupyter

click  
...wait  
click  
...wait  
click

New Upload

- Notebook
- Terminal
- Console
- New File
- New Folder

last month

File Edit View Run Kernel Settings Help Trusted

+ ✂ 📄 📁 ▶ ⏪ ⏩ Code ...

```
[1]:  
import numpy  
print(numpy)
```

<module 'numpy' from '/opt/conda/lib/python3.11/site-packages/numpy/\_init\_.py'>

WOW

# Containerized Scientific Applications

# Singularity with GPU

- Containers should be built with CUDA version compatible with local GPUs (CUDA  $\geq$  11)
- Just add the `--nv` flag to your singularity command

Many repositories on Docker Hub have GPU-ready images. Search for images with “gpu” in tags

The nvidia cloud also provides GPU-ready images. See:  
[hprc.tamu.edu/kb/Software/Singularity/Examples/#nvidia-gpu-cloud](https://hprc.tamu.edu/kb/Software/Singularity/Examples/#nvidia-gpu-cloud)

# NVIDIA Container Registry Example

**NVIDIA NGC | CATALOG** Welcome Guest

Catalog > Containers > PyTorch

## PyTorch

Accelerated with **NVIDIA**

**23.09-py3** Get Container Deploy to Vertex AI

Copy the latest tag's image path below:

```
nvcr.io/nvidia/pytorch:23.09-py3
```

**warning: do not attempt**

```
singularity pull docker://nvcr.io/nvidia/pytorch:23.09-py3
```

# PyTorch GPU Exercise

Image file: `pytorch_23.09-py3.sif`

*from* `docker://nvcr.io/nvidia/pytorch:23.09-py3`

*Located at* `/scratch/training/singularity/`

From the login node: (all on one line)

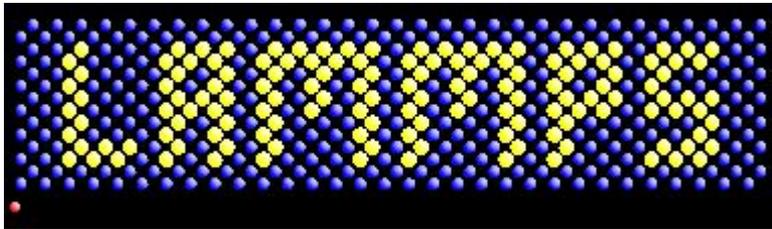
```
srun --mem=4G --time=60 --ntasks=1  --reservation=containers  
--gres=gpu:1 --partition=gpu --pty bash -i
```

From the compute node: (all on one line)

```
singularity exec --nv pytorch_23.09-py3.sif  
python3 -c "import torch;  
print(torch.cuda.device_count())"
```

# LAMMPS Molecular Dynamics on GPUs

- LAMMPS is a classical MD code
- <https://www.lammps.org/> has a cool animated logo.
- NVIDIA provides GPU-ready container images for lammps.  
<https://catalog.ngc.nvidia.com/orgs/hpc/containers/lammps>



# LAMMPS on H100 GPUs

- *This specific build works with H100 GPUs*

The screenshot shows the NVIDIA NGC Catalog interface for LAMMPS containers. The browser address bar displays `catalog.ngc.nvidia.com/orgs/hpc/containers/lammps/tags`. The page header includes the NVIDIA NGC | CATALOG logo and a 'Welcome Guest' message. The breadcrumb trail is 'Catalog > Containers > LAMMPS'. The main heading is 'LAMMPS' with a 'Get Container' button. Below the heading are tabs for 'Overview', 'Tags', 'Layers', 'Security Scanning', and 'Related Collections'. A search bar for tags is present. A highlighted container entry is shown with the following details:

Container Name	Repository	Created	Size	Architectures
patch_15Jun2023	<code>nvcr.io/hpc/lammps:patch_15Jun2023</code>	08/09/2023 11:34 AM	561.38 MB	2 Architectures

# LAMMPS on GPUs

Image file: `lammops-nv-patch_15Jun2023.sif`

*from* `docker://nvcr.io/hpc/lammops:patch_15Jun2023`

*Located at* `/scratch/training/singularity/`

From the login node: (all on one line)

```
srun --mem=4G --time=60 --ntasks=1  --gres=gpu:1 --partition=gpu --pty bash -i
```

following along live? add:  
`--reservation=containers`

From the compute node:

```
cd /scratch/training/singularity
```

(all on one line):

```
singularity run --nv lammops-nv-patch_15Jun2023.sif  
bash benchmark.sh
```

# Acknowledgements

This work was supported by

- the National Science Foundation (NSF), award numbers:
  - 2112356 - ACES - Accelerating Computing for Emerging Sciences
  - 1925764 - SWEETER - SouthWest Expertise in Expanding, Training, Education and Research
  - 2019129 - FASTER - Fostering Accelerated Scientific Transformations, Education, and Research
- Staff and students at Texas A&M High-Performance Research Computing.
- ACCESS CCEP pilot program, Tier-II



# ByteBoost 2026



***Do you want to learn how to accelerate your research through cutting-edge technologies?***

- 6 webinars and a week-long in-person workshop (travel, lodging, and food covered)
- Get access to 5 NSF-funded test-bed clusters
- Open to all U.S.-based researchers or educators

**More information  
and registration  
details here**





**HIGH PERFORMANCE  
RESEARCH COMPUTING**  
TEXAS A&M UNIVERSITY

<https://hprc.tamu.edu>

HPRC Helpdesk:

help@hprc.tamu.edu

Phone: 979-845-0219

*Take our short course survey!*



[https://u.tamu.edu/hprc\\_shortcourse\\_survey](https://u.tamu.edu/hprc_shortcourse_survey)

HPRC Survey

[https://u.tamu.edu/hprc\\_shortcourse\\_survey](https://u.tamu.edu/hprc_shortcourse_survey)

Help us help you. Please include details in your request for support, such as, Cluster (ACES, FASTER, Grace, Launch), NetID (UserID), Job information (JobID(s), Location of your jobfile, input/output files, Application, Module(s) loaded, Error messages, etc), and Steps you have taken, so we can reproduce the problem.



High Performance Research Computing | NSF award #2112356