

E4S: The Extreme-scale Scientific Software Stack Release 25.06

Release 25.06 notes
June 6, 2025



HPSF
HIGH PERFORMANCE
SOFTWARE FOUNDATION

High Performance Software Foundation
E4S Team



U.S. DEPARTMENT OF
ENERGY

Office of
Science

E4S 25.06: What's New?

- E4S includes 130+ HPC packages on ARM, x86_64, and ppc64le platforms.
- E4S improves support for a cross-platform AI/ML software stack including packages like NVIDIA BioNeMo™, NVIDIA NeMo™, Vllm, HuggingFace CLI, TensorFlow, PyTorch, Google.generativeai (Gemini API), OpenAI (API), TorchBraid, Pandas, Scikit-Learn, JAX, OpenCV, and LBANN with support for GPUs.
- Support for new architecture: NVIDIA Blackwell (sm_120).
- Updates to language and runtime: Chapel with support for AMD and NVIDIA GPUs
- New tools: libCEED
- Applications include: CP2K, DeaII, FFTX, GROMACS, LAMMPS, Nek500, Nekbone, NWChem, OpenFOAM, WarpX, WRF, Quantum Espresso, and Xyce
- E4S DocPortal updated with AI/ML tools.
- CUDA upgraded to 12.8 (aarch64, x86_64), ROCm upgraded to 6.3.3, oneAPI upgraded to 2025.1.
- Adaptive Computing's Heidi AI/HPC Cloud on demand data center (ODDC) web-based platform for multi-user, multi-node ParaTools Pro for E4S™ images on AWS, Azure, Google Cloud, and OCI Marketplace with NVIDIA GPUs with VNC based remote desktop and Torque (qsub) and SLURM (sbatch) for multi-node execution:

- <https://adaptivecomputing.com/> and <https://paratoolspro.com>



E4S: Extreme-scale Scientific Software Stack



<https://e4s.io>

About E4S

- E4S is an **ecosystem for science** and a community effort to provide open-source software packages for developing, deploying and running scientific applications on HPC platforms.
- E4S has built a comprehensive, coherent software stack that enables application developers to productively develop highly parallel applications that effectively target diverse exascale architectures.
- E4S provides a curated, Spack based software distribution of 130+ HPC (OpenFOAM, Gromacs, Nek5000, LAMMPS), EDA (e.g., Xyce), and AI/ML packages (e.g., NVIDIA NeMo™, NVIDIA BioNeMo™, Vllm, HuggingFace CLI, TensorFlow, PyTorch, OpenCV, TorchBraid, Scikit-Learn, Pandas, JAX, LBANN with support for GPUs).
- Base images and full featured containers (with GPU support) and DOE LLVM containers.
- Commercial support for E4S through ParaTools, Inc. for installation, maintaining an issue tracker, and ECP AD engagement.
- E4S for commercial clouds: Adaptive Computing's ODDC with ParaTools Pro for E4S™ image for **AWS, GCP, Azure, OCI**.
- With E4S Spack binary build caches, E4S supports both bare-metal and containerized deployment for GPU based platforms.
 - x86_64, ppc64le (IBM Power 10), aarch64 (ARM64) with support for CPUs and GPUs from NVIDIA, AMD, and Intel
 - Container images on DockerHub and E4S website of pre-built binaries of ECP ST products.
- e4s-chain-spack.sh to chain two Spack instances allows us to install new packages in home directory and use other tools.
- e4s-cl container launch tool allows binary distribution of applications by swapping MPI in the containerized app w/ system MPI.
- e4s-alc is an à la carte tool to customize container images by adding system and Spack packages to an existing image.
- E4S 25.06 released on June 6, 2025: https://e4s.io/talks/E4S_25.06.pdf



E4S Download from <https://e4s.io>



The screenshot shows the E4S website homepage. The browser's address bar displays <https://e4s.io>. The main header features the title "E4S: An HPC-AI Software Ecosystem for Science" in a large, white, serif font against a dark, textured background. Below the title is a navigation bar with links: HOME, USING E4S, JOINING E4S, ABOUT, CONTACT, EVENTS, PACKAGES, and DOWNLOAD. The PACKAGES and DOWNLOAD links are highlighted in orange. The main content area has a white background and contains three paragraphs of text. The first paragraph describes E4S as a community effort to provide an open-source software ecosystem for science. The second paragraph details the support for developing, deploying, and running scientific applications on HPC and AI platforms, mentioning sponsorship by the US Department of Energy (DOE) Office of Advanced Scientific Computing Research and use at the US Department of Defense, US National Science Foundation, and other federal agencies. The third paragraph highlights the provision of from-source builds, containers, and pre-installed versions of a broad collection of HPC and AI software packages, specifically mentioning the E4S 25.06 release announcement. The PESO logo is visible in the bottom left corner of the screenshot.

E4S: An HPC-AI Software Ecosystem for Science

HOME USING E4S JOINING E4S ABOUT CONTACT EVENTS PACKAGES DOWNLOAD

E4S is a community effort to provide and support an open-source software ecosystem for science. E4S provides a curated collection of scientific libraries and tools (packages) that form the foundation for hundreds of the world's most advanced scientific applications.

E4S packages support developing, deploying and running scientific applications on high-performance computing (HPC) and AI platforms sponsored by the **US Department of Energy (DOE) Office of Advanced Scientific Computing Research**. E4S is also used as a foundation for applications on leadership-class computing systems at the **US Department of Defense, US National Science Foundation, and other federal agencies**. It is used on numerous high-performance computing systems at universities and at collaborating international organizations.

E4S provides from-source builds, containers, and pre-installed versions of a **broad collection of HPC and AI software packages (E4S 25.06 release announcement)**. E4S includes contributions from many organizations, including national laboratories, universities, and industry. E4S is one of the key legacies of the **US Exascale Computing Project (ECP)**, a collaborative effort of the US Department of Energy Office of Advanced Scientific Computing Research and the National Nuclear Security Administration.

 PESO

E4S Container Download from https://e4s.io



The screenshot shows the E4S Container Download page. The browser address bar displays <https://e4s.io/download.html>. The page has a navigation bar with links: HOME, USING E4S, JOINING E4S, ABOUT, CONTACT, EVENTS, PACKAGES, and DOWNLOAD. The main heading is "Acquiring E4S Containers". Below this, there is a paragraph explaining the current E4S container offerings, including Docker and Singularity images for various architectures. A link to the E4S Docker Hub is provided. The page is divided into two main sections: "Container Releases" and "From source with Spack". The "Container Releases" section lists various download options, with "Docker Downloads - CUDA" highlighted by a blue box. The "From source with Spack" section includes a link to the Spack Project and a paragraph explaining that Spack contains packages for all products listed in the E4S 25.06 Full Release category.

Acquiring E4S Containers

The current E4S container offerings include Docker and Singularity images capable of running on X86_64, PPC64LE, and AARCH64 architectures. Our full E4S Release images are based on Ubuntu 22.04 (x86_64, aarch64, ppc64le). In addition to offering a full E4S image containing a comprehensive selection of E4S software released on a bi-annual cycle, we also offer a set of minimal base images suitable for use in Continuous Integration (CI) pipelines where Spack is used to build packages.

Docker images are available on the [E4S Docker Hub](#).

Please see the [E4S 25.06 Release Notes](#).

Container Releases

- Docker Downloads - CPU only
- Docker Downloads - CUDA**
- Docker Downloads - ROCm
- Docker Downloads - OneAPI
- Singularity x86_64 Download - CPU only
- Singularity x86_64 Download - CUDA 80
- Singularity x86_64 Download - CUDA 90
- Singularity x86_64 Download - CUDA 120
- Singularity ppc64le Download - CUDA 70
- Singularity aarch64 Download - CPU only
- Singularity aarch64 Download - CUDA 75
- Singularity aarch64 Download - CUDA 80
- Singularity aarch64 Download - CUDA 90
- Singularity x86_64 Download - ROCm gfx942
- Singularity x86_64 Download - ROCm gfx90a
- Singularity x86_64 Download - ROCm gfx908
- Singularity x86_64 Download - OneAPI
- OVA Download

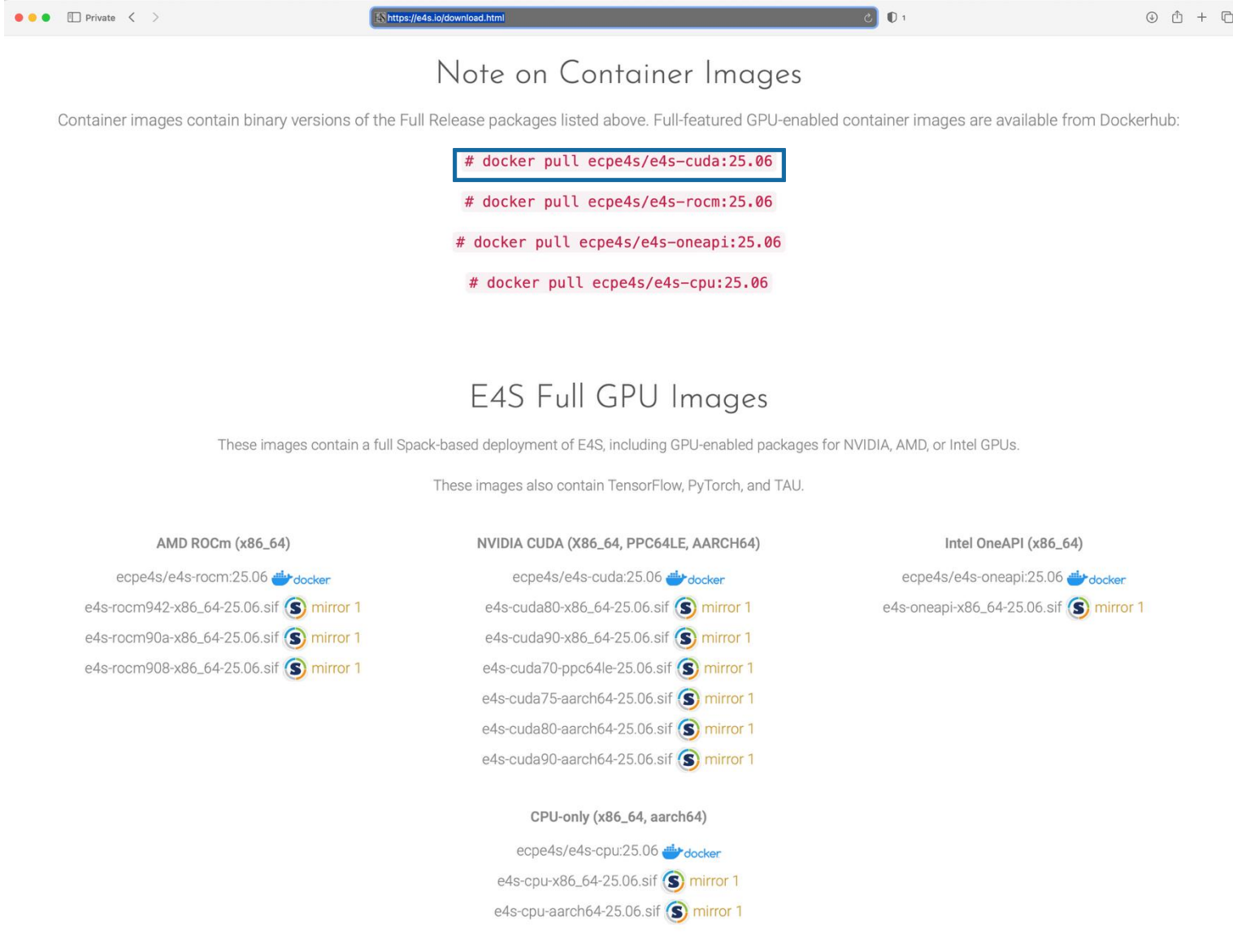
From source with Spack

[Visit the Spack Project](#)

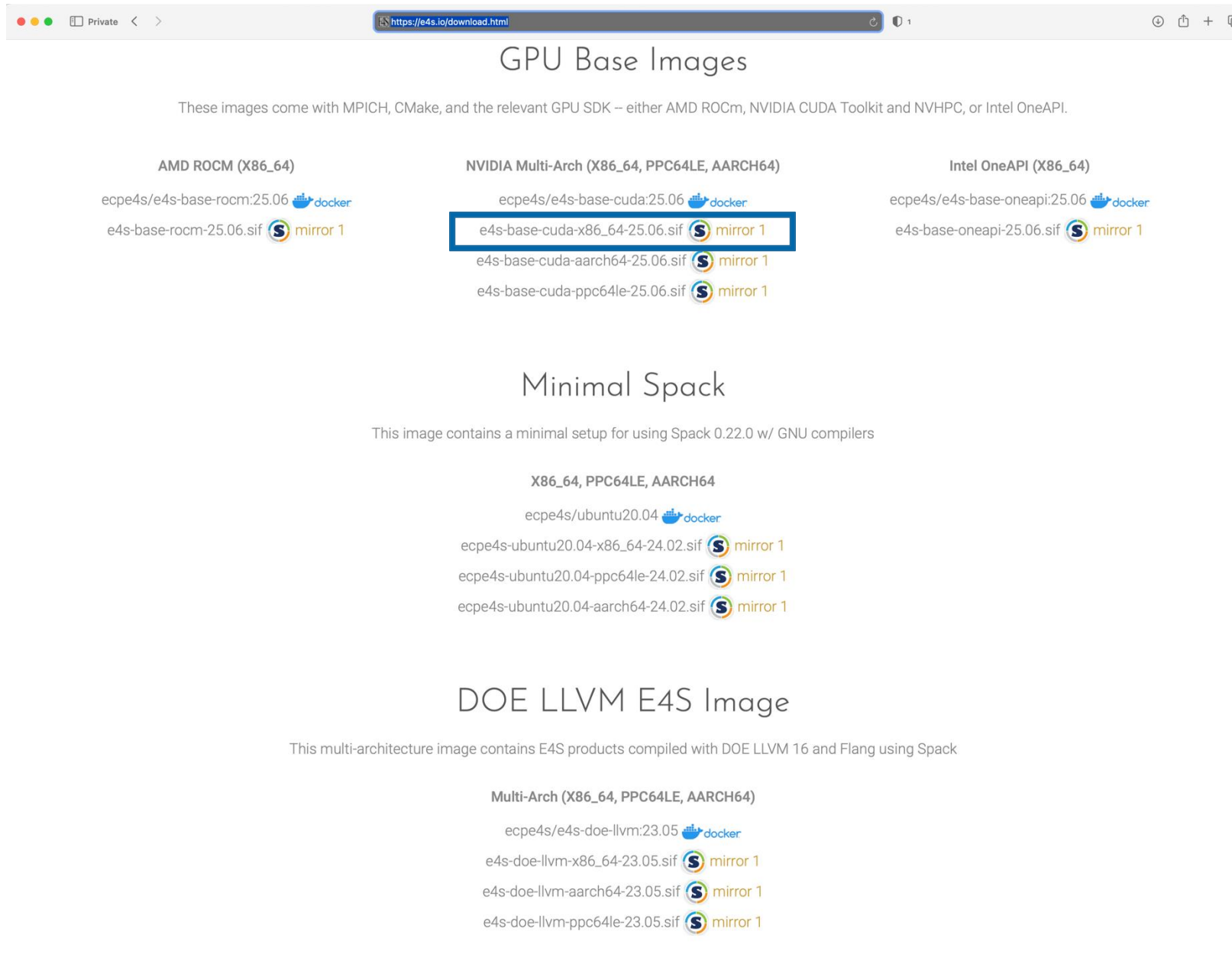
Spack contains packages for all of the products listed in the E4S 25.06 Full Release category (see above Release Notes). General instructions for building software with Spack can be found at the Spack website. Questions concerning building those packages are deferred to the associated package development team.

- Separate full featured Singularity images for 3 GPU architectures
- GPU full featured images for
 - x86_64 (Intel, AMD, NVIDIA)
 - ppc64le (NVIDIA)
 - aarch64 (NVIDIA)
- Full featured images available on Dockerhub
- 130+ products on 3 architectures

Download E4S 25.06 GPU Container Images: AMD, Intel, and NVIDIA











E4S base container images allow users to customize their containers



The screenshot shows a web browser window with the URL <https://e4s.io/download.html>. The page is titled "GPU Base Images" and contains three main sections: "AMD ROCm (X86_64)", "NVIDIA Multi-Arch (X86_64, PPC64LE, AARCH64)", and "Intel OneAPI (X86_64)". Each section lists Docker and Singularity image tags. The "NVIDIA Multi-Arch" section is highlighted with a blue box around the "e4s-base-cuda-x86_64-25.06.sif" tag. Below this, there is a "Minimal Spack" section and a "DOE LLVM E4S Image" section, both listing Docker and Singularity image tags.





GPU Base Images

These images come with MPICH, CMake, and the relevant GPU SDK -- either AMD ROCm, NVIDIA CUDA Toolkit and NVHPC, or Intel OneAPI.

AMD ROCm (X86_64)	NVIDIA Multi-Arch (X86_64, PPC64LE, AARCH64)	Intel OneAPI (X86_64)
ecpe4s/e4s-base-rocm:25.06 	ecpe4s/e4s-base-cuda:25.06 	ecpe4s/e4s-base-oneapi:25.06 
e4s-base-rocm-25.06.sif 	e4s-base-cuda-x86_64-25.06.sif 	e4s-base-oneapi-25.06.sif 
	e4s-base-cuda-aarch64-25.06.sif 	
	e4s-base-cuda-ppc64le-25.06.sif 	





Minimal Spack

This image contains a minimal setup for using Spack 0.22.0 w/ GNU compilers

X86_64, PPC64LE, AARCH64
ecpe4s/ubuntu20.04 
ecpe4s-ubuntu20.04-x86_64-24.02.sif 
ecpe4s-ubuntu20.04-ppc64le-24.02.sif 
ecpe4s-ubuntu20.04-aarch64-24.02.sif 

DOE LLVM E4S Image

This multi-architecture image contains E4S products compiled with DOE LLVM 16 and Flang using Spack


Multi-Arch (X86_64, PPC64LE, AARCH64)
ecpe4s/e4s-doe-llvm:23.05 
e4s-doe-llvm-x86_64-23.05.sif 
e4s-doe-llvm-aarch64-23.05.sif 
e4s-doe-llvm-ppc64le-23.05.sif 

- Intel oneAPI
- AMD ROCm
- NVIDIA CUDA


E4S Application Specific CI and minimal CI images

Application-Specific Continuous Integration Images


ecpe4s/solve-rocm6.3.0




ecpe4s/solve-cuda12.6.3-arm64




ecpe4s/solve-cuda12.6.3-amd64




ecpe4s/trilinos-sycl




ecpe4s/trilinos-ci-rocm6.2.1




ecpe4s/trilinos-ci-cuda12.2.2




ecpe4s/trilinos-ci-arm64-cuda12




ecpe4s/exawind-snapshot



ecpe4s/exago-cuda80




ecpe4s/exago-rocm90a




Minimal Continuous Integration Images


X86_64


ecpe4s/ubuntu22.04-runner-x86_64







ecpe4s/ubuntu20.04-runner-x86_64







ecpe4s/ubuntu18.04-runner-x86_64







ecpe4s/rhel8-runner-x86_64






ecpe4s/rhel7-runner-x86_64







PPC64LE


ecpe4s/ubuntu22.04-runner-ppc64le







ecpe4s/ubuntu20.04-runner-ppc64le







ecpe4s/ubuntu18.04-runner-ppc64le







ecpe4s/rhel8-runner-ppc64le






ecpe4s/rhel7-runner-ppc64le







AARCH64


ecpe4s/ubuntu22.04-runner-aarch64







ecpe4s/ubuntu20.04-runner-aarch64






ecpe4s/rhel8-runner-aarch64






Custom Images


ecpe4s/waggle-ml



ecpe4s/exawind-snapshot

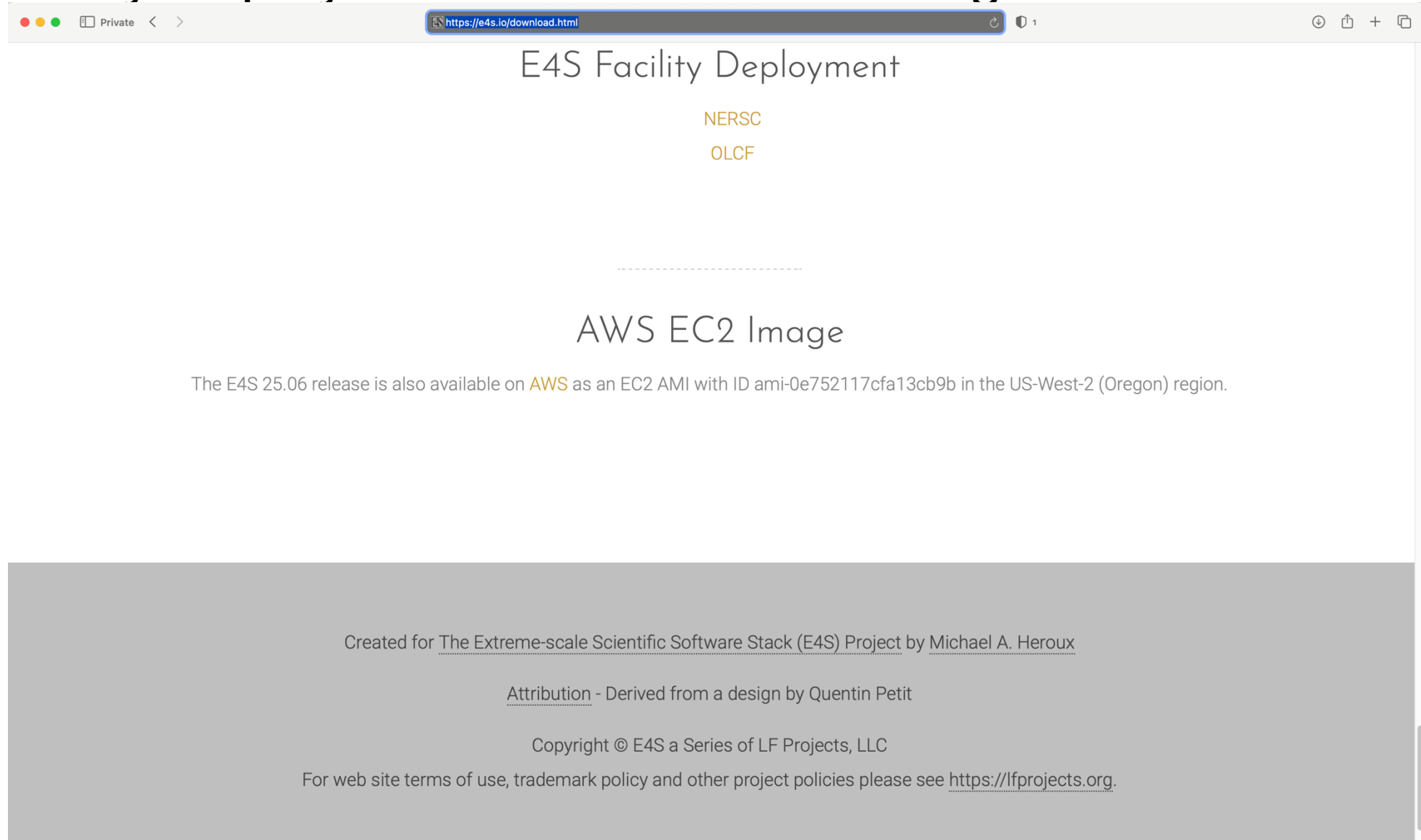


ecpe4s/superlu_sc





E4S Facility Deployment and AWS EC2 Image



E4S 25.06 Detailed Documentation for Bare-metal Installation



The screenshot shows the E4S website with the following content:

E4S: An HPC-AI Software Ecosystem for Science

HOME USING E4S JOINING E4S ABOUT CONTACT EVENTS PACKAGES DOWNLOAD

E4S offers multiple methods for deploying its comprehensive collection of HPC and AI software packages. Below is an overview of these deployment options. Additional documentation is available [here](#).



From-Source Builds

E4S utilizes [Spack](#), a flexible package manager, to facilitate building software directly from source. This approach allows users to customize builds according to their specific system architectures and requirements. Detailed instructions for manual installation are available in the [E4S Manual Installation Guide](#).



Containers

E4S offers containerized versions of its software stack, compatible with platforms such as Docker, Singularity, Shifter, and CharlieCloud. You can see the current list of E4S containers under the [Downloads page](#). These containers provide a consistent and portable environment for running HPC and AI applications across diverse systems. E4S provides containers from DockerHub and on cloud platforms, such as AWS, Azure, and Google Cloud.



Spack Build Cache

To expedite the installation process, E4S provides pre-built binaries through Spack build caches. These caches contain binaries for major operating systems and architectures, enabling users to install software without the need for local compilation. More information can be found on the [E4S About Page](#).



Cloud Options

- **Amazon Web Services (AWS):** E4S is available on AWS, allowing users to deploy the software stack on cloud-based infrastructure. This facilitates scalable and flexible computing resources for various workloads. Details about E4S on AWS can be found on the [E4S Home Page](#).
- **Google Cloud Platform (GCP):** Users can also deploy E4S on GCP, leveraging Google's cloud

E4S DocPortal updated with new applications and AI/ML tools

HOMEUSING E4SJOINING E4SABOUTCONTACTEVENTSPACKAGESDOWNLOAD

E4S Packages

Show 10 entries

Search: NVIDIA

	Name	Area	Description	Latest Doc Update
+	BIONEMO-FRAMEWORK	PMR	NVIDIA BioNeMo Framework is a comprehensive suite of programming tools, libraries, and models designed for computational drug discovery	2025-04-11 09:25:08
+	FPCHECKER	Tools	Floating point exception trapping for NVIDIA GPUs	2022-04-14 19:57:44
+	NEMO	AI	NVIDIA NeMo Framework is a scalable and cloud-native generative AI framework	2025-05-25 21:57:06

NameAreaDescription

Showing 1 to 3 of 3 entries (filtered from 144 total entries)

Previous1Next



E4S Tools: e4s-chain-spack.sh to customize software stack

```
sameer@mothra:~$ ls ~/images
e4s-cuda80-x86_64-25.06.sif
sameer@mothra:~$ singularity run --nv ~/images/e4s-cuda80-x86_64-25.06.sif
Singularity> /etc/e4s/e4s-chain-spack.sh ~/spack
Cloning into '/home/sameer/spack'...
remote: Enumerating objects: 686113, done.
remote: Counting objects: 100% (976/976), done.
remote: Compressing objects: 100% (463/463), done.
remote: Total 686113 (delta 772), reused 518 (delta 510), pack-reused 685137 (from 3)
Receiving objects: 100% (686113/686113), 230.82 MiB | 37.06 MiB/s, done.
Resolving deltas: 100% (326280/326280), done.
```

Specify location of downstream
Spack installation directory

```
-----
Configuration SUCCESS!
```

```
Downstream: /home/sameer/spack
Upstream: /spack
```

```
To use the downstream Spack instance, run the following command in your shell:
. /home/sameer/spack/share/spack/setup-env.sh
-----
```

Source downstream Spack's
setup-env.sh

```
Singularity> . /home/sameer/spack/share/spack/setup-env.sh
Singularity> spack find valgrind
==> Error: No package matches the query: valgrind
Singularity> spack install valgrind
[+] /usr/local/mpich/install/mpich (external mpich-4.2.3-47excoypwhfmx57rfs6reouvnninugcf)
[+] /usr (external glibc-2.35-a7drdl4tlx4bu3mzhor75pskvd3pdot6)
[+] /spack/opt/spack/linux-ubuntu22.04-x86_64_v3/gcc-11.4.0/gcc-runtime-11.4.0-f63c77kavzjtpmnhucd2oyfaxagwjzla
[+] /spack/opt/spack/linux-ubuntu22.04-x86_64_v3/gcc-11.4.0/boost-1.86.0-6qkv24gbidwxhllgah6jrkym5ev2cng5
[+] /spack/opt/spack/linux-ubuntu22.04-x86_64_v3/gcc-11.4.0/gmake-4.4.1-qp5blvcyuzgzhqsrp2ew6gq2nlos34b2
==> Installing valgrind-3.23.0-feuux36lsqp7quzmhmo4opbiadwpsars [6/6]
==> No binary for valgrind-3.23.0-feuux36lsqp7quzmhmo4opbiadwpsars found: installing from source
==> Fetching https://mirror.spack.io/_source-cache/archive/c5/c5c34a3380457b9b75606df890102e7df2c702b9420c2ebef9540f8b5d56264d.tar.bz2
==> Ran patch() for valgrind
==> valgrind: Executing phase: 'autoreconf'
==> valgrind: Executing phase: 'configure'
==> valgrind: Executing phase: 'build'
==> valgrind: Executing phase: 'install'
==> valgrind: Successfully installed valgrind-3.23.0-feuux36lsqp7quzmhmo4opbiadwpsars
Stage: 3.78s. Autoreconf: 0.01s. Configure: 48.56s. Build: 37.71s. Install: 2.97s. Post-install: 0.60s. Total: 1m 33.97s
[+] /home/sameer/spack/opt/spack/linux-ubuntu22.04-x86_64_v3/gcc-11.4.0/valgrind-3.23.0-feuux36lsqp7quzmhmo4opbiadwpsars
Singularity> spack load valgrind
Singularity> which valgrind
/home/sameer/spack/opt/spack/linux-ubuntu22.04-x86_64_v3/gcc-11.4.0/valgrind-3.23.0-feuux36lsqp7quzmhmo4opbiadwpsars/bin/valgrind
```

Install a new Spack package
in downstream Spack directory

Load new package (valgrind)
using spack load

E4S Tools: e4s-chain-spack.sh to customize software stack

```
Singularity> which valgrind
/home/sameer/spack/opt/spack/linux-ubuntu22.04-x86_64_v3/gcc-11.4.0/valgrind-3.23.0-feuwx36lsqp7quzmhmo4opbiadwpsars/bin/valgrind
Singularity> valgrind --help
usage: valgrind [options] prog-and-args
```

```
tool-selection option, with default in [ ]:
--tool=<name>          use the Valgrind tool named <name> [memcheck]
                        available tools are:
                        memcheck cachegrind callgrind helgrind drd
                        massif dhat lackey none exp-bbv

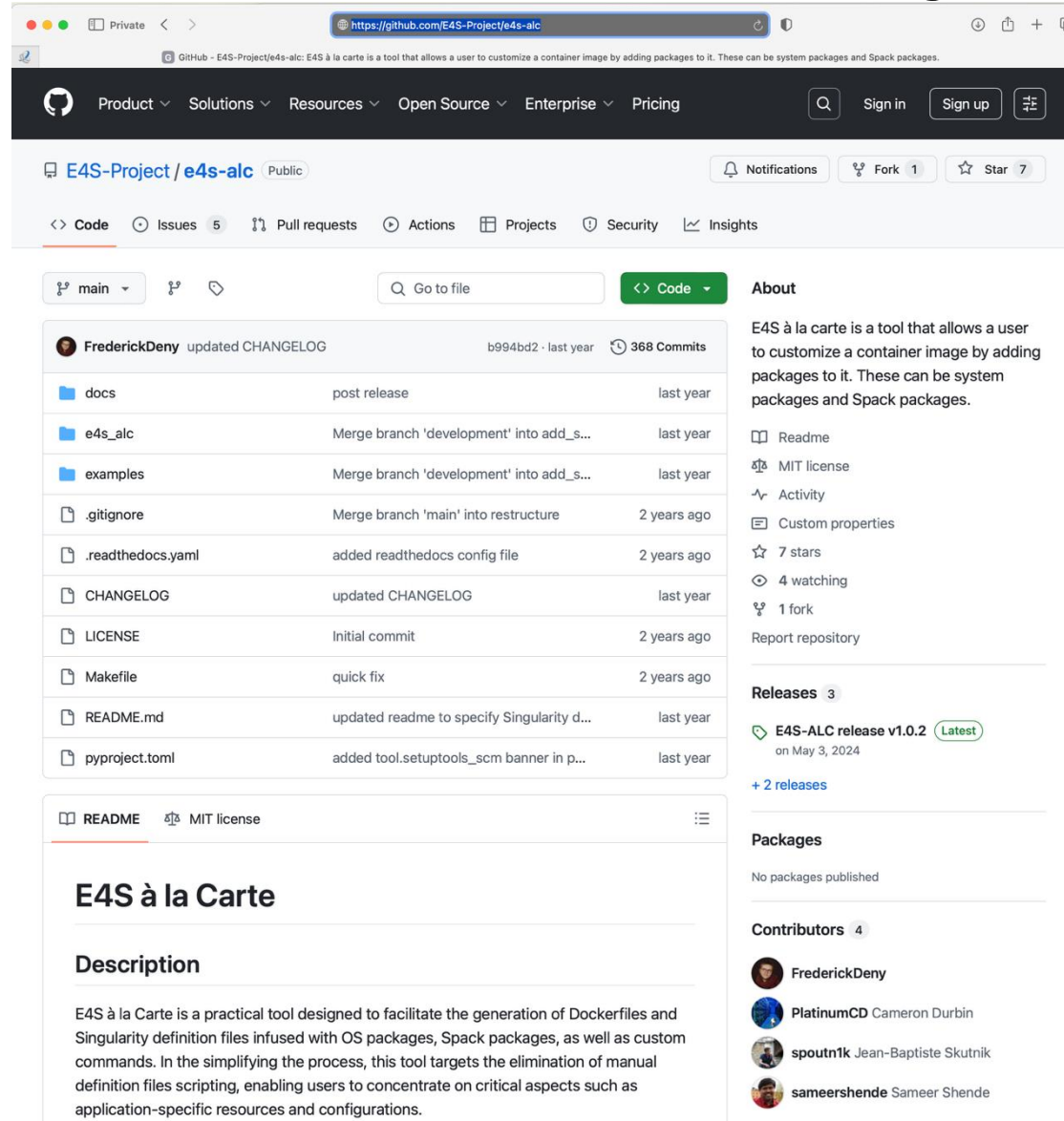
basic user options for all Valgrind tools, with defaults in [ ]:
-h --help              show this message
--help-debug           show this message, plus debugging options
--help-dyn-options     show the dynamically changeable options
--version              show version
-q --quiet             run silently; only print error msgs
-v --verbose           be more verbose -- show misc extra info
--trace-children=no|yes Valgrind-ise child processes (follow execve)? [no]
--trace-children-skip=patt1,patt2,... specifies a list of executables
                        that --trace-children=yes should not trace into
--trace-children-skip-by-arg=patt1,patt2,... same as --trace-children-skip=
                        but check the argv[] entries for children, rather
                        than the exe name, to make a follow/no-follow decision
--child-silent-after-fork=no|yes omit child output between fork & exec? [no]
--vgdb=no|yes|full     activate gdbserver? [yes]
                        full is slower but provides precise watchpoint/step
--vgdb-error=<number>  invoke gdbserver after <number> errors [999999999]
                        to get started quickly, use --vgdb-error=0
                        and follow the on-screen directions
--vgdb-stop-at=event1,event2,... invoke gdbserver for given events [none]
                        where event is one of:
                        startup exit abexit valgrindabexit all none
--track-fds=no|yes|all track open file descriptors? [no]
                        all includes reporting stdin, stdout and stderr
--time-stamp=no|yes     add timestamps to log messages? [no]
--log-fd=<number>       log messages to file descriptor [2=stderr]
--log-file=<file>       log messages to <file>
--log-socket=ipaddr:port log messages to socket ipaddr:port
--enable-debuginfod=no|yes query debuginfod servers for missing
                        debuginfo [yes]

user options for Valgrind tools that report errors:
--xml=yes              emit error output in XML (some tools only)
--xml-fd=<number>       XML output to file descriptor
--xml-file=<file>       XML output to <file>
--xml-socket=ipaddr:port XML output to socket ipaddr:port
--xml-user-comment=STR  copy STR verbatim into XML output
--demangle=no|yes       automatically demangle C++ names? [yes]
--num-callers=<number> show <number> callers in stack traces [12]
--error-limit=no|yes    stop showing new errors if too many? [yes]
--exit-on-first-error=no|yes exit code on the first error found? [no]
--error-exitcode=<number> exit code to return if errors found [0=disable]
--error-markers=<begin>,<end> add lines with begin/end markers before/after
                        each error output in plain text mode [none]
--show-error-list=no|yes|all show detected errors list and
                        suppression counts at exit [no].
                        all means to also print suppressed errors.
                        same as --show-error-list=yes
-s
```

Downstream Spack's package is loaded in your environment

e4s-chain-spack.sh helps customize the software stack using upstream /spack (read-only in the container) for package dependencies while installing a new package in the downstream Spack in your writable home directory.

E4S Tools: e4s-alc: Customize container images



GitHub - E4S-Project/e4s-alc: E4S à la carte is a tool that allows a user to customize a container image by adding packages to it. These can be system packages and Spack packages.

Product Solutions Resources Open Source Enterprise Pricing

E4S-Project / e4s-alc Public

Code Issues 5 Pull requests Actions Projects Security Insights

main Go to file Code

FrederickDeny updated CHANGELOG b994bd2 · last year 368 Commits

File	Commit Message	Time
docs	post release	last year
e4s_alc	Merge branch 'development' into add_s...	last year
examples	Merge branch 'development' into add_s...	last year
.gitignore	Merge branch 'main' into restructure	2 years ago
.readthedocs.yaml	added readthedocs config file	2 years ago
CHANGELOG	updated CHANGELOG	last year
LICENSE	Initial commit	2 years ago
Makefile	quick fix	2 years ago
README.md	updated readme to specify Singularity d...	last year
pyproject.toml	added tool.setuptools_scm banner in p...	last year

README MIT license

E4S à la Carte

Description

E4S à la Carte is a practical tool designed to facilitate the generation of Dockerfiles and Singularity definition files infused with OS packages, Spack packages, as well as custom commands. In the simplifying the process, this tool targets the elimination of manual definition files scripting, enabling users to concentrate on critical aspects such as application-specific resources and configurations.

About

E4S à la carte is a tool that allows a user to customize a container image by adding packages to it. These can be system packages and Spack packages.

Readme MIT license Activity Custom properties 7 stars 4 watching 1 fork Report repository

Releases 3

E4S-ALC release v1.0.2 Latest on May 3, 2024 + 2 releases

Packages

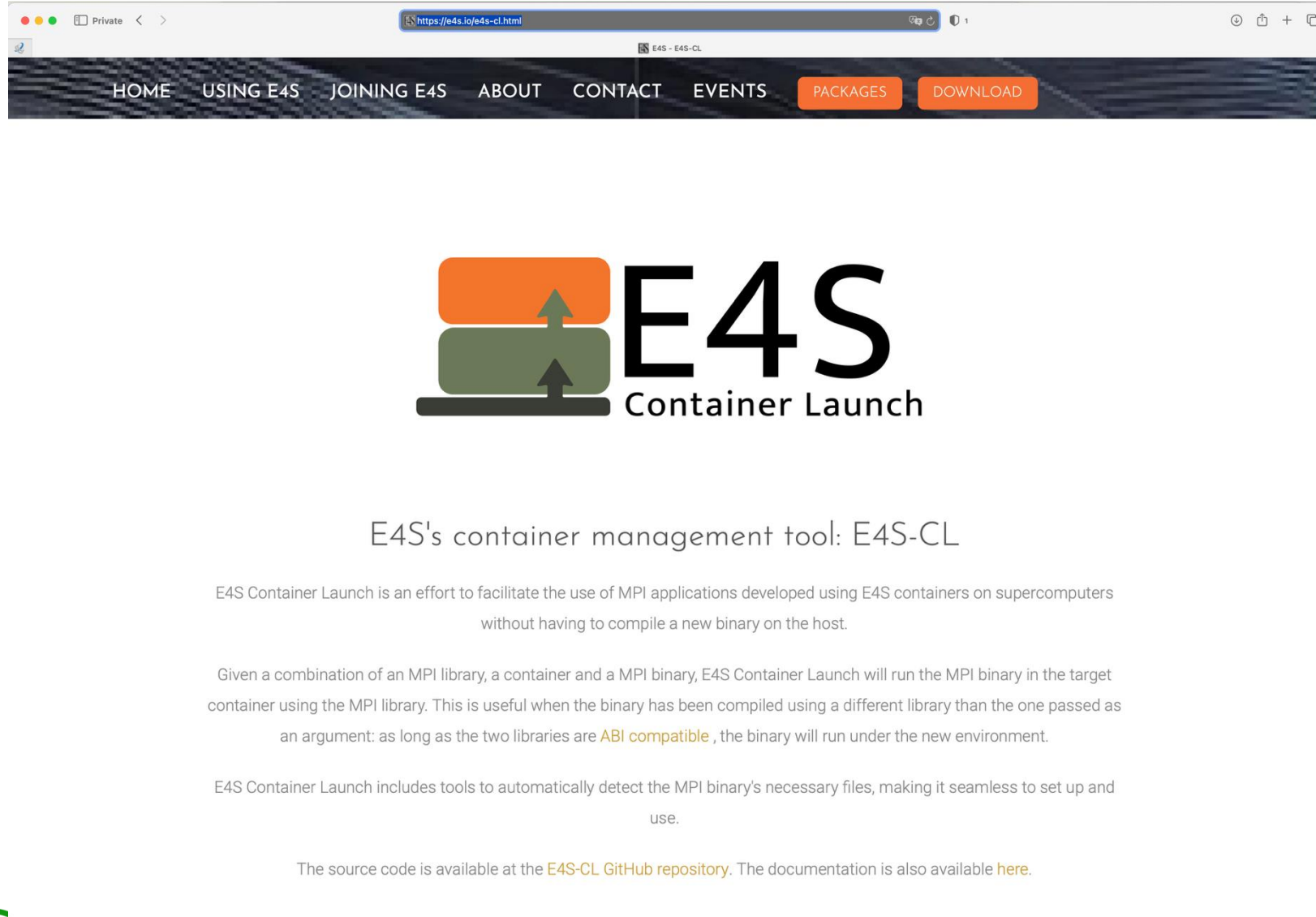
No packages published

Contributors 4

- FrederickDeny
- PlatinumCD Cameron Durbin
- spoutn1k Jean-Baptiste Skutnik
- sameershende Sameer Shende

- Add new system packages
- Add new Spack packages
- Add new tarballs
- Customize the container image
- Start with a base image
- Add packages
- Create a new container image!

E4S Tools: e4s-cl: Container Launch tool for MPI applications



- Distribute your MPI application as a binary with an E4S image
- While deploying on a system substitute the embedded containerized MPI in application with the system/vendor MPI
- Use inter-node network interfaces efficiently for near native performance!

e4s-cl: A tool to simplify the launch of MPI jobs in E4S containers

The screenshot displays the GitHub repository for `E4S-Project/e4s-cl`. The repository is public and has 4 forks and 16 stars. The commit history shows a series of updates, including adding workflows, assets, and fixing issues. The README section, titled "E4S Container Launcher", describes the tool as a way to simplify running MPI applications in E4S containers. It mentions that the project is built upon the [MPICH ABI Compatibility Initiative](#) and CEA's [Wi4MPI](#). A link to the documentation is provided. The "Usage" section is also visible.



<https://github.com/E4S-Project/e4s-cl>

e4s-cl: A tool to simplify the launch of MPI jobs in E4S containers

- E4S containers support replacement of MPI libraries using MPICH ABI compatibility layer and Wi4MPI [CEA] for OpenMPI replacement.
- Applications binaries built using E4S can be launched with Singularity using MPI library substitution for efficient inter-node communications.
- e4s-cl is a new tool that simplifies the launch and MPI replacement.

- e4s-cl init --backend [singularity|shifter|docker] --image <file> --source <startup_cmds.sh>
 - e4s-cl mpirun -np <N> <command>

- Usage:

```
% e4s-cl init --backend singularity --image ~/images/e4s-gpu-x86.sif --source ~/source.sh
% cat ~/source.sh
. /spack/share/spack/setup-env.sh
spack load trilinos+cuda cuda_arch=90
% e4s-cl mpirun -np 4 ./a.out
```



E4S on Commercial Cloud Platforms: ParaTools Pro for E4S™

* Acknowledgment:
Supported by
DOE SBIR Phase I and II
DE-SC0022502



U.S. DEPARTMENT OF
ENERGY

Office of
Science

<https://paratoolspro.com> and <https://www.energy.gov/technologytransitions/sbirsttr>

[PRODUCTS](#)
[SERVICES](#)
[SOLUTIONS](#)
[TUTORIALS](#)
[TRAINING](#)
[NEWS](#)
[MORE..](#)

PARATOOLS PRO FOR E4S™

ParaTools Pro for E4S™ — the Extreme-scale Scientific Software Stack — hardened for commercial clouds and supported by ParaTools, Inc. provides a platform for developing and deploying HPC and AI/ML applications. It features a performant remote desktop environment (based on VNC) on the login node and compute nodes interconnected by a low-latency, high bandwidth network adapter based on high-speed network interface cards such as AWS Elastic Fabric Adapter (EFA). ParaTools Pro for E4S™ features a suite of over 100 HPC tools built using the Spack package manager and the proprietary MVAPICH MPI tuned for commercial cloud platforms. It features ready to use HPC applications (such as OpenFOAM, LAMMPS, CP2K, Xyce, and Quantum Espresso) as well as AI/ML tools based on Python (such as NVIDIA NeMo™, TensorFlow, PyTorch, JAX, Horovod, Keras, OpenCV, SciKit Learn, and Pandas), supports Jupyter notebooks and the Codium IDE. New packages can be easily installed using Spack and pip and are accessible on the cluster compute and login nodes featuring NVIDIA GPUs. Job scheduling is supported using the Torque and SLURM batch schedulers. It may be used for developing the next generation of generative AI applications using a suite of Python tools and interfaces.

The Extreme-scale Scientific Software Stack (**E4S**) has built a unified computing environment for deployment of open-source projects. E4S was originally developed to provide a common software environment for the exascale leadership computing systems currently deployed at DOE National Laboratories across the US. ParaTools, Inc. is offering support for deploying and testing E4S products.

Adaptive AI Cloud with the On Demand Data Center and E4S

OpenFOAM is the free, open source CFD software developed primarily by OpenCFD Ltd since 2004. It has a large and active base user area of engineering and research from both commercial and academic organisations. OpenFOAM has a reputation for extensive high-fidelity flow modelling capabilities and its flexibility allows it to be adapted to many different types of problems. OpenFOAM is particularly valued for its ability to include customised numerical developments and extensions from the community, its adaptability to various hardware architectures, its extensibility through plug-in modules and its support for multi-scale simulations, mesh adaptation and parallelisation.

OpenFOAM is a leading tool for computational fluid dynamics (CFD) and is widely used in industry and academia. It is a powerful tool for simulating complex flows and is supported by a large community of users and developers. The software is designed to be flexible and adaptable, allowing users to tailor the code to their specific needs. OpenFOAM is a free and open source software project, which means that anyone can download, modify, and distribute the code without any restrictions. This makes it a popular choice for researchers and engineers who want to develop customised solutions for their specific problems.

Watch on YouTube

ParaTools Pro for E4S™ image with Adaptive Computing's ODDC

Deploy with adaptive.hpc.ai/ml-as-a-service


Deploy ParaTools Pro for E4S™ through **Adaptive Computing, Inc.'s adaptive.hpc.ai/ml-as-a-service** using a simple web interface to deploy on Amazon Web Services, Google Cloud Platform, Microsoft Azure, and Oracle Cloud Infrastructure.

Images for use with adaptive.hpc.ai/ml-as-a-service are available as Cloud Marketplace images:

Cloud Platform	ParaTools Pro for E4S™ Images
Amazon Web Services	ParaTools Pro for E4S™ on ODDC Node (AWS, x86-64) ParaTools Pro for E4S™ on ODDC Server (AWS, x86-64) ParaTools Pro for E4S™ on ODDC Node (AWS, arm64) ParaTools Pro for E4S™ on ODDC Server (AWS, arm64)
Google Cloud Platform	ParaTools Pro for E4S™ on ODDC (GCP, x86-64)
Microsoft Azure	ParaTools Pro for E4S™ on ODDC (Azure, x86-64)
Oracle Cloud Infrastructure	ParaTools Pro for E4S™ on ODDC (OCI, x86-64)

- ParaTools Pro for E4S™*
images in vendor marketplaces support:
 - AWS
 - Azure
 - Google Cloud (GCP)
 - Oracle Cloud Infrastructure (OCI)
- Supports SLURM and Torque for scheduling jobs on multi-node GPU accelerated nodes
- Shared GPU accelerated login node with a VNC based remote desktop
- Adaptive Computing's Heidi/ODDC
- AWS PCS and PC (x86, ARM64)
- Azure Cyclecloud
- Google GCluster

E4S on Adaptive Computing's Heidi AI/On Demand Data Center (ODDC)



The screenshot shows the Heidi AI website. At the top is the Heidi logo with the tagline 'Empowering Minds'. Below it is the 'Our Mission' section, stating Heidi AI's goal is to provide every student with access to their own personal supercomputer. The main section is 'Heidi for Grades K-12 & Higher Education', described as a 'Your Cloud-Based Personal AI Supercomputer for Grades K-12 & Higher Education'. It introduces cutting-edge technology to high school classrooms and higher education institutions. Below this is a section titled 'Cloud-Based Supercomputing for Education: HPC, AI, and STEM Solutions with Heidi'. This section is divided into three columns: 'How it works', 'Heidi Technology Stack', and 'ParaTools Pro for E4S™ - Extreme-scale Scientific Software Stack'. The 'How it works' column explains that Heidi brings the power of supercomputing to K-12 and higher education. The 'Heidi Technology Stack' column lists features like On-Demand Data Center (ODDC) for multi-cloud access, a comprehensive curriculum on building and deploying HPC & AI clusters, and scalable resources. The 'ParaTools Pro' column describes how it utilizes Heidi's infrastructure to provide students with an integrated suite of HPC & AI tools. At the bottom is the 'Heidi AI Excellence Scholarship' section, which states that beginning in 2026, each Heidi school district will be invited to submit one or more student created AI applications for a chance to be awarded a \$10,000 scholarship for each team member, up to \$50,000 per team.

Our Mission

Heidi AI's mission is to provide every student with access to their own personal supercomputer, ensuring that all students, regardless of their economic background, have the tools they need to succeed and reach their full potential.

Heidi for Grades K-12 & Higher Education

Your Cloud-Based Personal AI Supercomputer for Grades K-12 & Higher Education

Introducing cutting-edge technology to high school classrooms and higher education institutions can be daunting. Schools often lack access to high-performance computing (HPC) resources and AI infrastructure, and building or maintaining such infrastructure can be prohibitively expensive. Educators face challenges integrating computational tools into their curricula, from accessing preloaded datasets to finding software that bridges technical complexity with student-friendly learning environments. Heidi helps future-proof technology and will support quantum computing when perfected.

Heidi for Grades K-12 and Higher Education is a cloud-based, SaaS supercomputing platform designed to make HPC & AI accessible and affordable for educational institutions. Heidi leverages powerful infrastructure, preloaded datasets, and educational tools to help students simulate real-world phenomena, build AI applications, and explore advanced computational concepts. With partnerships that include leading scientific organizations, some funded by grants from the US Department of Energy, Heidi ensures educators and students alike are equipped with world-class resources to achieve hands-on learning in science, technology, engineering, and mathematics—all at an accessible per-student annual price.

Heidi's cost per student is less than the cost of a textbook. Heidi also integrates with on-premises data centers.

Cloud-Based Supercomputing for Education: HPC, AI, and STEM Solutions with Heidi

How it works	Heidi Technology Stack	ParaTools Pro for E4S™ - Extreme-scale Scientific Software Stack
Heidi brings the power of supercomputing to K-12 and higher education, offering an intuitive platform that makes HPC & AI accessible and effective for educators and students alike. By simplifying deployment and providing robust resources, Heidi enables hands-on learning.	Heidi empowers students and educators by providing an accessible, cloud-based supercomputing platform tailored to K-12 and higher education. It simplifies the integration of HPC & AI into classrooms with preloaded datasets, easy-to-use tools, and scalable infrastructure.	ParaTools Pro for E4S™ utilizes Heidi's robust infrastructure to provide students and educators with an integrated suite of HPC & AI tools designed specifically for educational environments. Assembled under a DOE (U.S. Department of Energy) grant and included in the Adaptive Computing Technology Stack, ParaTools Pro for E4S™ simplifies the deployment of complex applications and empowers students to explore cutting-edge technologies in a hands-on setting. With over 150 preloaded applications, including TensorFlow, PyTorch, and Keras, ParaTools Pro for E4S™ enables learners to seamlessly engage in HPC & AI, computational science, and STEM-focused challenges using Heidi's robust infrastructure.
<ul style="list-style-type: none">Heidi gives every student access to their own personal HPC & AI supercomputer in the cloud. Included with Heidi is the best practices curriculum of how to use Heidi, visual examples of 3D renderings from HPC & AI scientific applications, and over 150 open source HPC & AI applications.	<ul style="list-style-type: none">Features the On-Demand Data Center (ODDC) for multi-cloud access and automated infrastructure deployment.Includes a comprehensive curriculum on building and deploying HPC & AI clusters in the cloud.Scales resources effortlessly while ensuring secure and reliable access via cloud-based infrastructure.	<ul style="list-style-type: none">Supports specialized hardware, including GPUs and high-speed network adapters, on major cloud platforms such as AWS, Azure, GCP, and OCI.Integrates seamlessly with Slurm, Moab/Torque, and MPI environments for efficient workload management.Provides a unified experience across commercial cloud environments, ensuring flexibility and scalability.

Heidi AI Excellence Scholarship

Beginning in 2026, each Heidi school district will be invited to submit one or more student created AI applications for a chance to be awarded a \$10,000 scholarship for each team member, up to \$50,000 per team.

- ParaTools Pro for E4S™ images in commercial cloud marketplaces launched using Heidi
- Supports Torque for scheduling jobs on multi-node GPU accelerated nodes
- Shared GPU accelerated login node with a VNC based remote desktop

ParaTools Pro for E4S™ on Commercial Clouds: AWS Marketplace

The screenshot shows the AWS Marketplace search results for 'ParaTools Pro for E4S'. The search bar at the top contains the text 'ParaTools Pro for E4S'. The results are displayed in a list format, showing 8 results. The left sidebar contains filters for Categories, Delivery methods, Publisher, Pricing model, Operating system, Free trial, Contract type, Architecture, and Region. The main content area displays four results for ParaTools Pro for E4S, each with a 'Free Trial' badge and a description of the software stack and its capabilities.

ParaTools Pro for E4S (8 results) showing 1 - 8
Did you mean [paratool pre e4?](#) Sort By: Relevance

ParaTools Pro for E4S™: AI/ML & HPC Tools on ParallelCluster (arm64)
By [ParaTools Inc.](#) | Ver v2025.03.05.1047-pcluster-3.12.0-e4s-24.11-arm64
Free Trial
Starting from \$0.99 to \$0.99/hr for software + AWS usage fees
ParaTools Pro for E4S™ - the Extreme-scale Scientific Software Stack, E4S™ hardened for commercial clouds and supported by ParaTools, Inc. provides a platform for developing and deploying HPC and AI/ML applications. It features a performant remote desktop environment (based on DCV) on the login...

ParaTools Pro for E4S™: AI/ML & HPC Tools on ParallelCluster (x86)
By [ParaTools Inc.](#) | Ver v2025.03.24.1709-pcluster-3.12.0-e4s-24.11-amd64
Free Trial
Starting from \$0.99 to \$0.99/hr for software + AWS usage fees
ParaTools Pro for E4S™ - the Extreme-scale Scientific Software Stack, E4S™ hardened for commercial clouds and supported by ParaTools, Inc. provides a platform for developing and deploying HPC and AI/ML applications. It features a performant remote desktop environment (based on DCV) on the login...

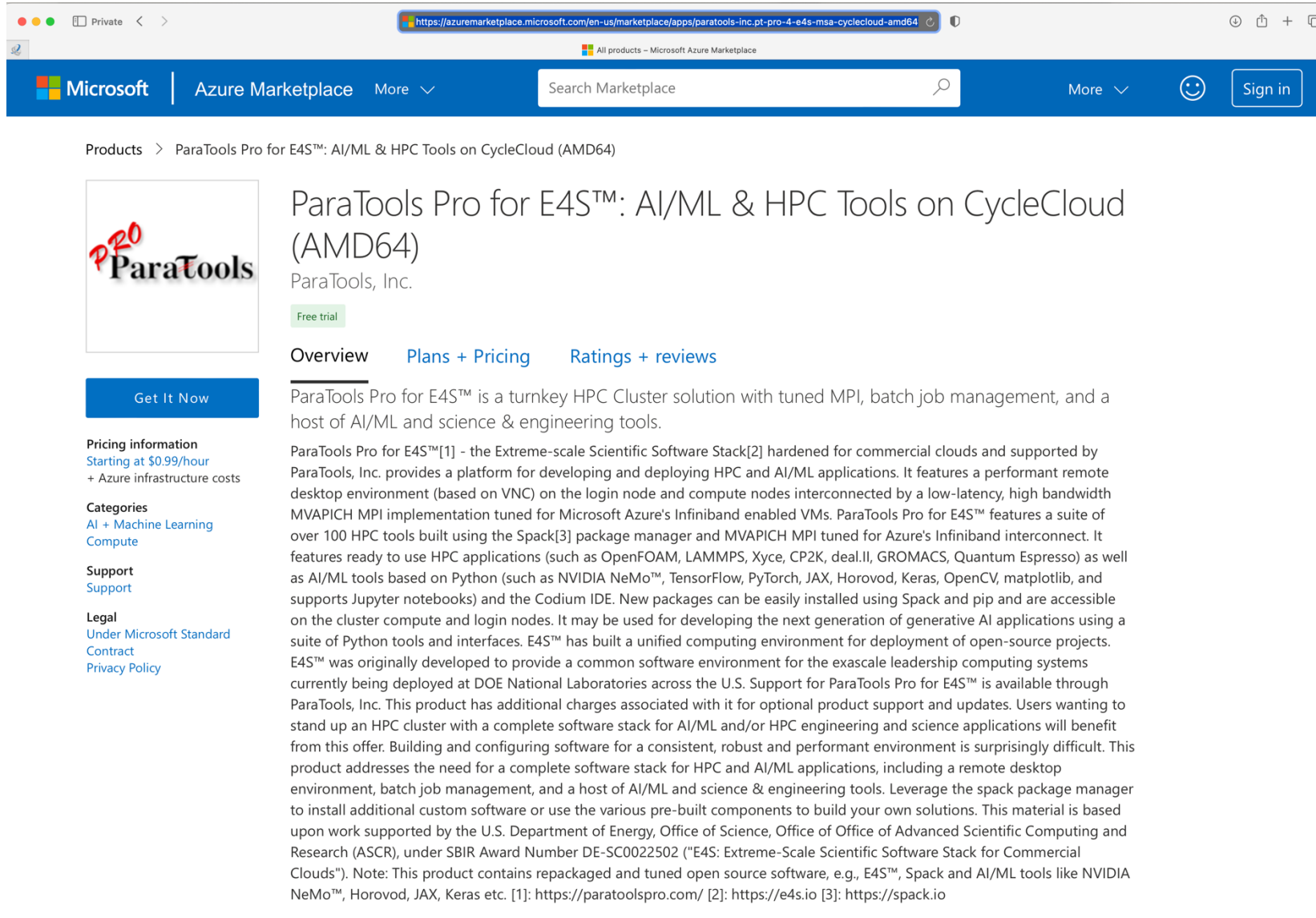
ParaTools Pro for E4S™: AI/ML & HPC Tools on AWS PCS (x86)
By [ParaTools Inc.](#) | Ver v2025.03.18.1119-pcs-1.1.1-e4s-24.11-amd64
Free Trial
Starting from \$0.10 to \$0.99/hr for software + AWS usage fees
ParaTools Pro for E4S™ - the Extreme-scale Scientific Software Stack, E4S™ hardened for commercial clouds and supported by ParaTools, Inc. provides a platform for developing and deploying HPC and AI/ML applications. It features a performant remote desktop environment (based on DCV) on the login...

ParaTools Pro for E4S™: AI/ML & HPC Tools on ODDC Node (x86)
By [ParaTools Inc.](#) | Ver v17423266-odd-v1.0-e4s-24.11-node-amd64
Free Trial
Starting from \$0.99 to \$0.99/hr for software + AWS usage fees
ParaTools Pro for E4S™ - the Extreme-scale Scientific Software Stack, E4S™ hardened for commercial clouds and supported by ParaTools, Inc. provides a platform for developing and deploying HPC and AI/ML applications. It features a performant remote desktop environment (based on VNC) on the login...

ParaTools Pro for E4S™
on AWS supports

- AWS Trainium and Inferentia custom AI hardware with NeuronX SDK
- AWS PCS and PC on x86_64 and aarch64 nodes
- NVIDIA GPUs
- SLURM (PCS and PC) and Torque (ODDC node/server)
- Also on AWS Marketplace in GovCloud (US East & West)
- Elastic Fabric Adapter (EFA)
- MVAPICH MPI
- [X-ScaleSolutions, LLC and The Ohio State University]

ParaTools Pro for E4S™ on Commercial Clouds: Azure Marketplace



Products > ParaTools Pro for E4S™: AI/ML & HPC Tools on CycleCloud (AMD64)

ParaTools Pro for E4S™: AI/ML & HPC Tools on CycleCloud (AMD64)

ParaTools, Inc.

Free trial

[Overview](#) [Plans + Pricing](#) [Ratings + reviews](#)

[Get It Now](#)

Pricing information
Starting at \$0.99/hour
+ Azure infrastructure costs

Categories
[AI + Machine Learning Compute](#)

Support
[Support](#)

Legal
[Under Microsoft Standard Contract](#)
[Privacy Policy](#)

ParaTools Pro for E4S™ is a turnkey HPC Cluster solution with tuned MPI, batch job management, and a host of AI/ML and science & engineering tools.

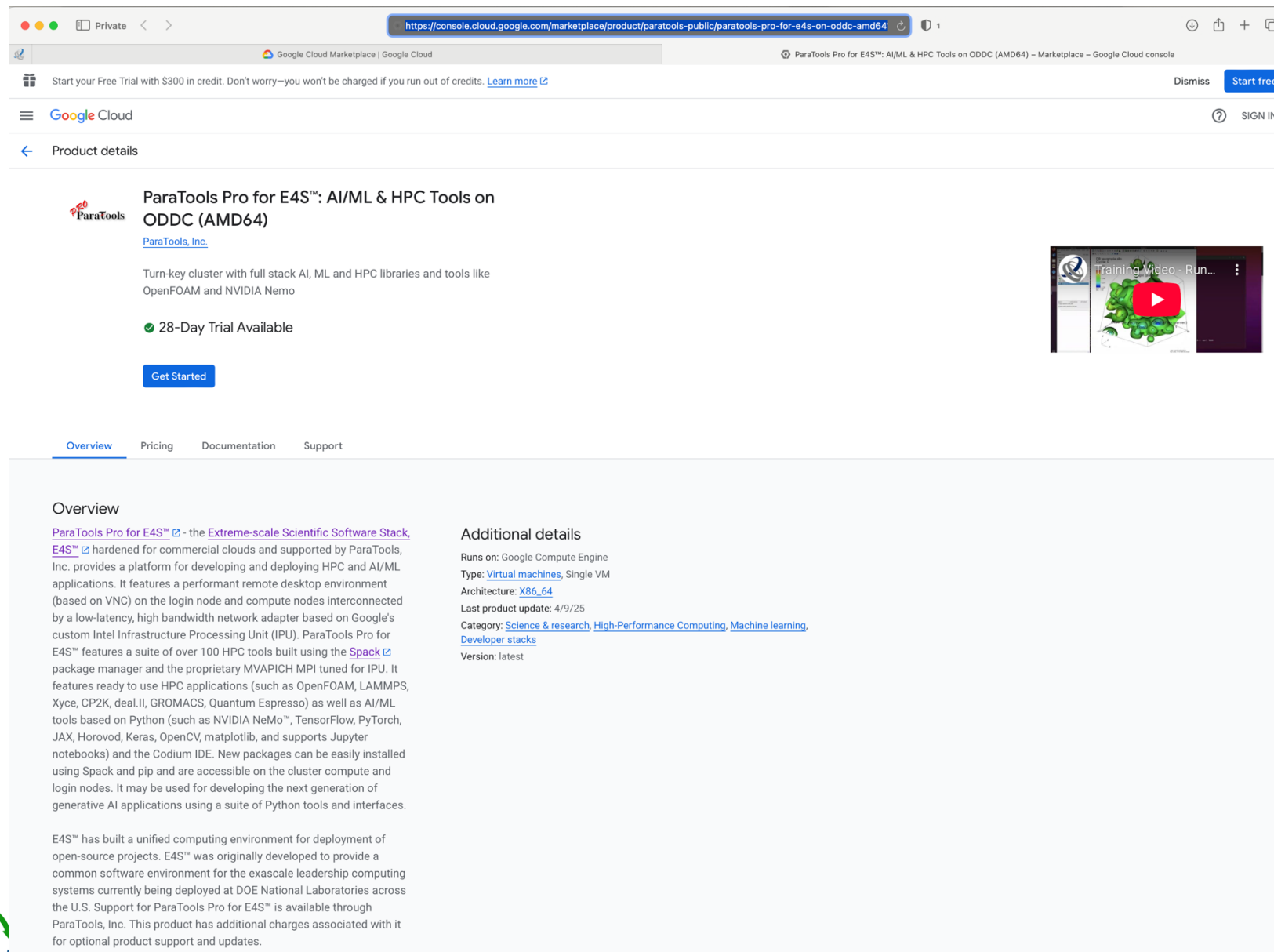
ParaTools Pro for E4S™[1] - the Extreme-scale Scientific Software Stack[2] hardened for commercial clouds and supported by ParaTools, Inc. provides a platform for developing and deploying HPC and AI/ML applications. It features a performant remote desktop environment (based on VNC) on the login node and compute nodes interconnected by a low-latency, high bandwidth MVAPICH MPI implementation tuned for Microsoft Azure's Infiniband enabled VMs. ParaTools Pro for E4S™ features a suite of over 100 HPC tools built using the Spack[3] package manager and MVAPICH MPI tuned for Azure's Infiniband interconnect. It features ready to use HPC applications (such as OpenFOAM, LAMMPS, Xyce, CP2K, deal.II, GROMACS, Quantum Espresso) as well as AI/ML tools based on Python (such as NVIDIA NeMo™, TensorFlow, PyTorch, JAX, Horovod, Keras, OpenCV, matplotlib, and supports Jupyter notebooks) and the Codium IDE. New packages can be easily installed using Spack and pip and are accessible on the cluster compute and login nodes. It may be used for developing the next generation of generative AI applications using a suite of Python tools and interfaces. E4S™ has built a unified computing environment for deployment of open-source projects. E4S™ was originally developed to provide a common software environment for the exascale leadership computing systems currently being deployed at DOE National Laboratories across the U.S. Support for ParaTools Pro for E4S™ is available through ParaTools, Inc. This product has additional charges associated with it for optional product support and updates. Users wanting to stand up an HPC cluster with a complete software stack for AI/ML and/or HPC engineering and science applications will benefit from this offer. Building and configuring software for a consistent, robust and performant environment is surprisingly difficult. This product addresses the need for a complete software stack for HPC and AI/ML applications, including a remote desktop environment, batch job management, and a host of AI/ML and science & engineering tools. Leverage the spack package manager to install additional custom software or use the various pre-built components to build your own solutions. This material is based upon work supported by the U.S. Department of Energy, Office of Office of Advanced Scientific Computing and Research (ASCR), under SBIR Award Number DE-SC0022502 ("E4S: Extreme-Scale Scientific Software Stack for Commercial Clouds"). Note: This product contains repackaged and tuned open source software, e.g., E4S™, Spack and AI/ML tools like NVIDIA NeMo™, Horovod, JAX, Keras etc. [1]: <https://paratoolspro.com/> [2]: <https://e4s.io> [3]: <https://spack.io>

ParaTools Pro for E4S™

on Azure Marketplace supports

- SLURM (Azure CycleCloud) and Torque schedulers (Adaptive Computing's ODDC)
- Support for Infiniband Network adapter

ParaTools Pro for E4S™ on Google Cloud Marketplace



Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

Google Cloud

Product details

ParaTools Pro for E4S™: AI/ML & HPC Tools on ODDC (AMD64)

[ParaTools, Inc.](#)

Turn-key cluster with full stack AI, ML and HPC libraries and tools like OpenFOAM and NVIDIA Nemo

✓ 28-Day Trial Available

[Get Started](#)

[Overview](#) [Pricing](#) [Documentation](#) [Support](#)

Overview

[ParaTools Pro for E4S™](#) - the [Extreme-scale Scientific Software Stack, E4S™](#) hardened for commercial clouds and supported by ParaTools, Inc. provides a platform for developing and deploying HPC and AI/ML applications. It features a performant remote desktop environment (based on VNC) on the login node and compute nodes interconnected by a low-latency, high bandwidth network adapter based on Google's custom Intel Infrastructure Processing Unit (IPU). ParaTools Pro for E4S™ features a suite of over 100 HPC tools built using the [Spack](#) package manager and the proprietary MVAICH MPI tuned for IPU. It features ready to use HPC applications (such as OpenFOAM, LAMMPS, Xyce, CP2K, deal.II, GROMACS, Quantum Espresso) as well as AI/ML tools based on Python (such as NVIDIA NeMo™, TensorFlow, PyTorch, JAX, Horovod, Keras, OpenCV, matplotlib, and supports Jupyter notebooks) and the Codium IDE. New packages can be easily installed using Spack and pip and are accessible on the cluster compute and login nodes. It may be used for developing the next generation of generative AI applications using a suite of Python tools and interfaces.

E4S™ has built a unified computing environment for deployment of open-source projects. E4S™ was originally developed to provide a common software environment for the exascale leadership computing systems currently being deployed at DOE National Laboratories across the U.S. Support for ParaTools Pro for E4S™ is available through ParaTools, Inc. This product has additional charges associated with it for optional product support and updates.

Additional details

Runs on: Google Compute Engine
Type: [Virtual machines](#), Single VM
Architecture: [X86_64](#)
Last product update: 4/9/25
Category: [Science & research](#), [High-Performance Computing](#), [Machine learning](#), [Developer stacks](#)
Version: latest

ParaTools Pro for E4S™

Google Cloud Marketplace supports

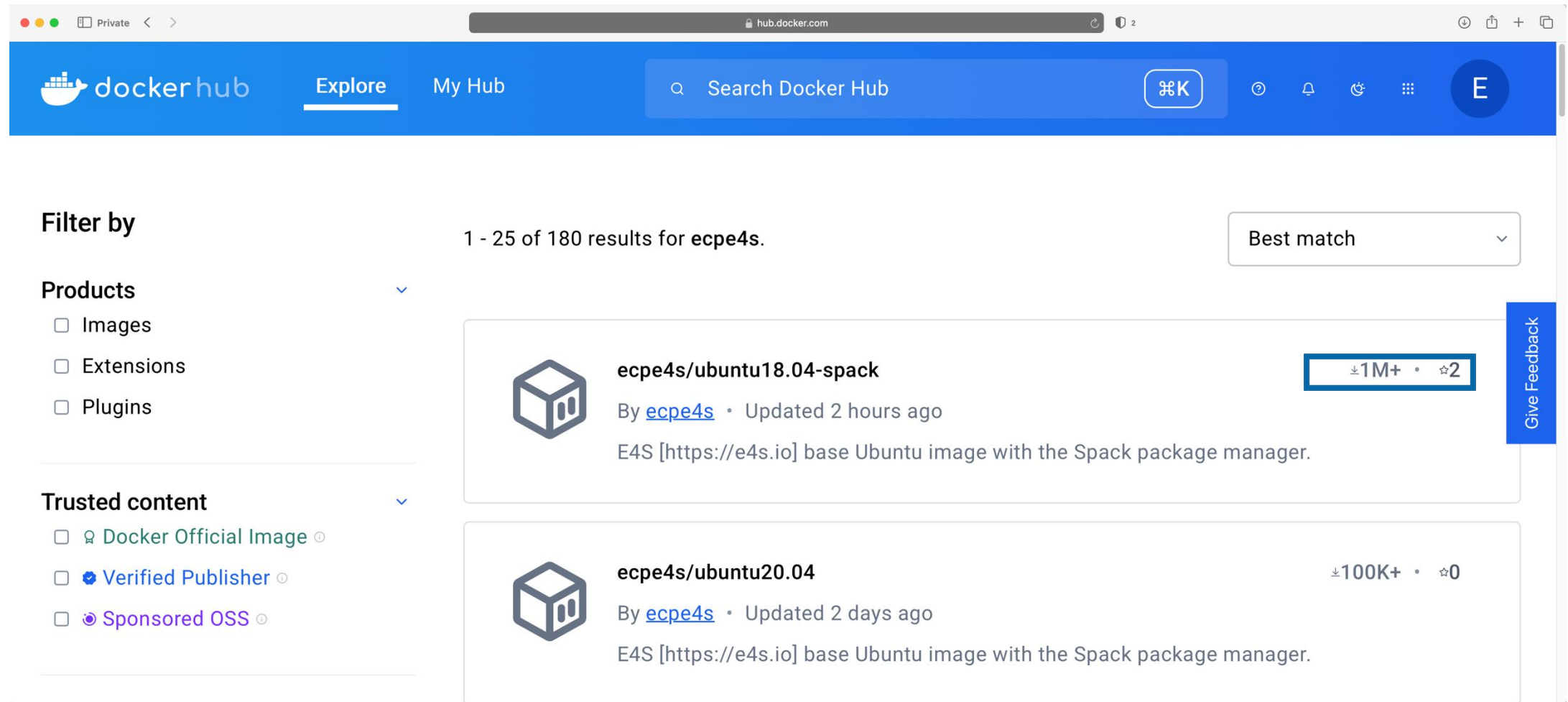
- SLURM (GCluster) and Torque schedulers (Adaptive Computing's ODDC)
- Support for Google IPU network adapters

ParaTools Pro for E4S™ on Oracle Cloud Marketplace

The screenshot displays the Oracle Cloud Marketplace interface for the 'ParaTools Pro for E4S™: AI/ML & HPC Tools on ODDC (AMD64)' application. The header includes the Oracle logo and navigation links. The main content area features the app's icon, title, and a 'Get App' button. Below this, a description states it is a 'Turn-key cluster with full stack AI, ML and HPC libraries and tools like OpenFOAM and NVIDIA Nemo™'. The pricing is listed as 'Software Price Instance per hour USD: 0.99'. A detailed overview section follows, explaining that ParaTools Pro is a hardened Extreme-scale Scientific Software Stack using HPC Cloud ODDC, supporting over 100 HPC and AI/ML tools. It highlights features like a performant remote desktop environment for GUI tools and a suite of Python tools for developing generative AI applications. The overview also mentions that the software is based on work supported by the U.S. Department of Energy and is available through ParaTools, Inc. for optional product support and updates.

ParaTools Pro for E4S™
for Oracle Cloud Infrastructure
(OCI) Marketplace
supports Torque (ODDC) and
RDMA over Converged Ethernet
(RoCE) network adapters and
GPUs on login and compute nodes

E4S 25.06 Release on DockerHub: Base images



The screenshot shows the DockerHub website interface. The top navigation bar is blue with the DockerHub logo, 'Explore' and 'My Hub' tabs, a search bar, and user profile icons. The main content area displays search results for 'ecpe4s'. On the left, there are filters for 'Products' (Images, Extensions, Plugins) and 'Trusted content' (Docker Official Image, Verified Publisher, Sponsored OSS). The search results show two items:

- ecpe4s/ubuntu18.04-spack**: By [ecpe4s](#) • Updated 2 hours ago. E4S [https://e4s.io] base Ubuntu image with the Spack package manager. Download: 1M+, Stars: 2.
- ecpe4s/ubuntu20.04**: By [ecpe4s](#) • Updated 2 days ago. E4S [https://e4s.io] base Ubuntu image with the Spack package manager. Download: 100K+, Stars: 0.

A 'Give Feedback' button is visible on the right side of the results area.

E4S 25.06 Release on DockerHub: Full Featured CUDA image

ecpe4s
Docker Free Team

Members

Teams

Repositories

Settings

General

Default privacy

Registry Access

Deactivate org

Billing

Usage

Pulls

Storage

Repositories / e4s-cuda / General

ecpe4s/e4s-cuda

Last pushed 1 day ago · Repository size: 1.6 TB

E4S [https://e4s.io] image with CUDA 12.8. Includes NVIDIA NeMo(TM), NVIDIA BioNeMo(TM) frameworks.

MACHINE LEARNING & AI DEVELOPER TOOLS OPERATING SYSTEMS

General Tags Image Management Permissions Webhooks Settings

Tags

This repository contains 40 tag(s).

Tag	OS	Health	Pulled	Pushed
latest		N/A	1 day	1 day
25.06-cuda90		N/A	1 day	2 days
25.06-cuda80		N/A	4 days	2 days
25.06-cuda75		N/A	4 days	2 days
25.06		N/A	1 day	4 days

See all

Repository overview

E4S, an HPSF project, is an HPC-AI software ecosystem for science. It's a curated, Spack based collection of scientific libraries and tools that form the foundation of some of the world's most advanced scientific applications. E4S curates the largest single collection of open-source GPU-enabled libraries and tools for scientific applications, including support for equation-based modeling and simulation, and AI for science. All E4s public-domain software is thoroughly tested for interoperability and portability to multiple computing architectures and will continue to be enhanced and expanded to address architectural changes and emerging new architectures. While E4S supports many products and distributions, users can confidently select any subset of functionality. We build and test the whole so you can select what you need. The key features of E4S 25.06 support a timely expansion of the ecosystem's AI portfolio and include:

- Support for the NVIDIA Blackwell GPU architecture with CUDA v12.8.
- NVIDIA NeMoTM Framework v2.3.0rc0, a comprehensive framework for building, customizing, deploying, and maintaining generative AI models. It includes support for large language models (LLMs), video models, vision language models (VLMs), and speech AI.

Using 0 of 0 private repositories. Get more

Docker commands

To push a new tag to this repository:

docker push ecpe4s/e4s-cuda:tagname

Public view

Scout health score

Not available

We couldn't assess the health of your most recently pushed tag. Enable Docker Scout to gain insights into your images and enhance your application's security. Learn more

View on Scout

buildcloud

Build with Docker Build Cloud

Accelerate image build times with access to cloud-based builders and shared cache.

Docker Build Cloud executes builds on optimally-dimensioned cloud infrastructure with dedicated per-organization isolation.

Get faster builds through shared caching across your team, native multi-platform support, and encrypted data transfer - all without managing infrastructure.

Go to Docker Build Cloud



https://e4s.io

E4S 25.06 Release on DockerHub: Full Featured ROCm image

ecpe4s
Docker Free Team

Members

Teams

Repositories

Settings

General

Default privacy

Registry Access

Deactivate org

Billing

Usage

Pulls

Storage

Repositories / e4s-rocm / General

ecpe4s/e4s-rocm

Last pushed 2 days ago • Repository size: 438.5 GB

E4S [https://e4s.io], a software ecosystem for science, is a collection of HPC-AI tools with ROCm.

LANGUAGES & FRAMEWORKS

MACHINE LEARNING & AI

DEVELOPER TOOLS

General

Tags

Image Management

Permissions

Webhooks

Settings

Tags

This repository contains 28 tag(s).

Tag	OS	Health	Pulled	Pushed
latest		N/A	2 days	2 days
25.06		N/A	2 days	2 days
25.06-gfx908		N/A	2 days	2 days
25.06-gfx90a		N/A	2 days	2 days
25.06-gfx942		N/A	2 days	2 days

See all

Repository overview

E4S, an HPSF project, is an HPC-AI software ecosystem for science. It's a curated, Spack based collection of scientific libraries and tools that form the foundation of some of the world's most advanced scientific applications. E4S curates the largest single collection of open-source GPU-enabled libraries and tools for scientific applications, including support for equation-based modeling and simulation, and AI for science. All E4s public-domain software is thoroughly tested for interoperability and portability to multiple computing architectures and will continue to be enhanced and expanded to address architectural changes and emerging new architectures. While E4S supports many products and distributions, users can confidently select any subset of functionality. We build and test the whole so you can select what you need. E4S tools updated in release 25.06 include e4s-chain-spack.sh to chain two Spack instances to allow users to install new packages while leveraging pre-installed E4S packages and dependencies, e4s-alc to customize containers from base container images, and e4s-cl to support binary distribution of MPI applications by substituting the MPI library embedded in the containerized application with the system/vendor MPI for near native performance. E4S 25.06 has separate CPU and GPU containers for AMD GPUs optimizing the size of the containers for each GPU architecture.

Using 0 of 0 private repositories. [Get more](#)

Docker commands

To push a new tag to this repository:

docker push ecpe4s/e4s-rocm:tagname

Public view

Scout health score

Not available

We couldn't assess the health of your most recently pushed tag. Enable Docker Scout to gain insights into your images and enhance your application's security. [Learn more](#)

View on Scout

buildcloud

Build with Docker Build Cloud

Accelerate image build times with access to cloud-based builders and shared cache.

Docker Build Cloud executes builds on optimally-dimensioned cloud infrastructure with dedicated per-organization isolation.

Get faster builds through shared caching across your team, native multi-platform support, and encrypted data transfer - all without managing infrastructure.

Go to Docker Build Cloud



https://e4s.io

E4S 25.06 Release on DockerHub: Full Featured oneAPI image

The screenshot shows the DockerHub repository page for `ecpe4s/e4s-oneapi`. The repository is owned by the `ecpe4s` Docker Free Team. It is a public repository with a size of 205.3 GB. The description states: "E4S [https://e4s.io], an ecosystem for science, provides HPC-AI tools with Intel(R) oneAPI." The repository is categorized under "LANGUAGES & FRAMEWORKS", "MACHINE LEARNING & AI", and "DEVELOPER TOOLS".

The "Tags" section shows 11 tags. The table below lists the tags and their details:

Tag	OS	Health	Pulled	Pushed
latest	linux/amd64	N/A	2 days	2 days
25.06	linux/amd64	N/A	2 days	2 days
24.11	linux/amd64	N/A	about 2 months	6 months
24.05	linux/amd64	N/A	5 months	about 1 year
24.02	linux/amd64	N/A	7 months	over 1 year

The "Repository overview" section states: "E4S, an HPSF project, is an HPC-AI software ecosystem for science. It's a curated, Spack based collection of scientific libraries and tools that form the foundation of some of the world's most advanced scientific applications. E4S curates the largest single collection of open-source GPU-enabled libraries and tools for scientific applications, including support for equation-based modeling and simulation, and AI for science. All E4s public-domain software is thoroughly tested for interoperability and portability to multiple computing architectures and will continue to be enhanced and expanded to address architectural changes and emerging new architectures. While E4S supports many products and distributions, users can confidently select any subset of functionality. We build and test the whole so you can select what you need."

E4S includes support for PyTorch optimized for Intel GPUs, Intel 25.1 C, C++, and Fortran compilers with Intel MPI. Other applications of note in E4S 25.06 include CP2K, Dealii, FFTX, GROMACS, LAMMPS, Nek5000, Nekbone, NWChem, OpenFOAM, WarpX, WRF, Quantum Espresso, and Xyce, with support for GPUs where available. VSCodium provides the integrated development environment, Jupyter notebook provides a web-based Python notebook interface. 3D graphics tools include ParaView, VisIt, and TAU. E4S tools updated in release 25.06 include `e4s-chain-spack.sh` to chain two Spack instances to allow

The "Scout health score" section shows a shield icon and the text: "Not available. We couldn't assess the health of your most recently pushed tag. Enable Docker Scout to gain insights into your images and enhance your application's security. [Learn more](#)"

The "buildcloud" section promotes Docker Build Cloud, stating: "Accelerate image build times with access to cloud-based builders and shared cache. Docker Build Cloud executes builds on optimally-dimensioned cloud infrastructure with dedicated per-organization isolation. Get faster builds through shared caching across your team, native multi-platform support, and encrypted data transfer - all without managing infrastructure. [Go to Docker Build Cloud](#)



<https://e4s.io>

E4S 25.06 image for NVIDIA H100 GPU on x86_64

```
$ singularity run --nv e4s-cuda90-x86_64-25.06.sif
Singularity> ls /opt/demo/e4s-cloud-examples/
bionemo      cuda      julia-cuda  lammps      nalu        openfoam    pytorch     superlu-dist-cpu  vllm
clean-all.sh fetch-all.sh julia-mpi    machine-learning nemo        osu-benchmarks pytorch-gpu  tau              xyce
CoMD          horovod    jupyter-notebook matmult      nemo-speech_to_text petsc-cpu    pytorch-image-classifier tensorflow
containers   jax        laghos      mpi-procname neuronx      petsc-cuda   qe           visit
Singularity> ls /opt/demo/e4s-cloud-examples/machine-learning/
clean.sh gemini openai perplexity pytorch tensorflow
Singularity> ls /opt/demo/e4s-cloud-examples/vllm
gradio_openai_chatbot_webserver.py llama2_template.jinja README.md run.sh run-smaller.sh
Singularity> python
Python 3.10.12 (main, Feb 4 2025, 14:57:36) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import nemo
>>> import bionemo
>>> import torch
>>> import openai
>>> import google.generativeai
>>> import huggingface_hub
>>> import jax
>>> import pandas
>>> import cv2
>>> import sklearn
>>> import mpi4py
>>> import matplotlib
>>> import seaborn
>>> import plotly
>>> import vllm
>>> vllm.__version__
'0.8.3.dev0+g25f560a62.d20250520'
>>> nemo.__version__
'2.3.0rc0'
>>> tensorflow.__version__
'2.19.0'
>>> torch.__version__
'2.6.0'
>>> torch.cuda.get_arch_list()
['sm_80', 'sm_90', 'sm_120']
>>>
```

E4S 25.06 image for NVIDIA GPUs (x86_64)

Singularity> spack find

-- linux-ubuntu22.04-x86_64_v3 / gcc@11.4.0 -----

abseil-cpp@20240722.0	expat@2.7.0	libceed@0.12.0	openssl@3.4.1	py-urllib3@2.1.0
adiak@0.4.1	fftw@3.3.10	libdwarf@0.11.0	otf2@3.0.3	py-wheel@0.45.1
adios2@2.10.2	fftw@3.3.10	libedit@3.1-20240808	papi@7.1.0	python@3.10.12
amrex@25.03	fftx@1.2.0	libevent@2.1.12	papi@7.1.0	python-venv@1.0
arborx@1.5	findutils@4.10.0	libfabric@1.22.0	parmetis@4.0.3	raja@2024.07.0
arpack-ng@3.9.1	flecsi@2.3.2	libffi@3.4.6	parmetis@4.0.3	raja@2024.07.0
asio@1.32.0	flex@2.6.3	libiberty@2.41	parsec@3.0.2209	re2c@3.1
autoconf@2.72	flux-core@0.67.0	libiconv@1.17	pcre2@10.44	readline@8.2
automake@1.16.5	fmt@11.1.4	libidn2@2.3.7	pdt@3.25.2	sed@4.9
axom@0.10.1	fmt@11.1.4	libint@2.9.0	perl@5.40.0	slate@2024.10.29
bc@1.07.1	gasnet@2024.5.0	libmd@1.1.0	perl@5.40.0	slepc@3.22.2
berkeley-db@18.1.40	gcc-runtime@11.4.0	libmonitor@2023.03.15	perl-data-dumper@2.173	snappy@1.2.1
binutils@2.43.1	gdbm@1.23	libpciaccess@0.17	petsc@3.22.4	spiral-package-fftx@1.3.0
bison@3.8.2	gettext@0.23.1	libpng@1.6.39	petsc@3.22.4	spiral-package-jit@1.1.0
blaspp@2024.10.26	ginkgo@1.9.0	libsigsegv@2.14	pigz@2.8	spiral-package-mpi@1.1.0
blt@0.7.0	git@2.48.1	libsodium@1.0.20	pkgconf@2.3.0	spiral-package-simt@1.1.0
blt@0.7.0	glibc@2.35	libtool@2.4.7	protobuf@3.28.2	spiral-software@8.5.1
boost@1.86.0	gmake@4.4.1	libunistring@1.2	protobuf@3.29.3	sqlite@3.46.0
boost@1.86.0	gmp@6.3.0	libunwind@1.8.1	py-calver@2022.6.26	strumpack@8.0.0
boost@1.86.0	gperftools@2.15	libxc@7.0.0	py-certifi@2023.7.22	sundials@7.2.1
boost@1.86.0	gromacs@2024.4	libxcrypt@4.4.38	py-cffi@1.17.1	superlu-dist@9.1.0
bricks@2023.08.25	hdf5@1.8.23	libxml2@2.13.5	py-charset-normalizer@3.3.0	superlu-dist@9.1.0
butterflypack@3.2.0	hdf5@1.14.5	libyaml@0.1.7	py-cython@3.0.11	sz@2.1.12.5
bzip2@1.0.8	hdf5@1.14.5	libyaml@0.2.5	py-editables@0.5	tar@1.35
c-blosc2@2.15.1	heffte@2.4.1	libzmq@4.3.5	py-flit-core@3.10.1	tasmanian@8.1
ca-certificates-mozilla@2025-02-25	hpctoolkit@2024.01.1	lizard@2.0	py-fypp@3.1	tau@2.34.1
cabana@0.7.0	hpcviewer@2025.01	llvm@19.1.7	py-hatchling@1.25.0	texinfo@7.1
caliper@2.12.1	hpx@1.10.0	lua@5.3.6	py-idna@3.4	trilinos@16.1.0
camp@2024.07.0	hwloc@2.11.1	lua@5.4.6	py-meson-python@0.16.0	umpire@2024.07.0
camp@2024.07.0	hwloc@2.11.1	lua-luaposix@36.1	py-numpy@2.2.4	umpire@2024.07.0
chai@2024.07.0	hypre@2.32.0	lz4@1.10.0	py-packaging@24.2	umpire@2024.07.0
chapel@2.4.0	hypre@2.32.0	lzo@2.10	py-pathspect@0.11.1	unzip@6.0
	icu4c@74.2	m4@1.4.19	py-pip@24.3.1	upcxx@2023.9.0



<https://e4s.io>

E4S 25.06 image for NVIDIA GPUs (x86_64)

```
Singularity> spack find +cuda
-- linux-ubuntu22.04-x86_64_v3 / gcc@11.4.0 -----
adios2@2.10.2      camp@2024.07.0    gromacs@2024.4    kokkos@4.6.01     nvcomp@2.2.0      strumpack@8.0.0    upcxx@2023.9.0
amrex@25.03        chai@2024.07.0    heffte@2.4.1      kokkos-kernels@4.6.01  papi@7.1.0        sundials@7.2.1     vtk-m@2.2.0
arborx@1.5         chapel@2.4.0      hpctoolkit@2024.01.1  lammps@20240829.1    parsec@3.0.2209    superlu-dist@9.1.0  zfp@1.0.0
axom@0.10.1        cp2k@2025.1       hpx@1.10.0        lapackpp@2024.10.26  petsc@3.22.4       tasmanian@8.1      tau@2.34.1
blaspp@2024.10.26  cusz@0.14.0      hwloc@2.11.1      legion@24.12.0      raja@2024.07.0     trilinos@16.1.0    umpire@2024.07.0
bricks@2023.08.25  fftx@1.2.0       hypre@2.32.0      libceed@0.12.0      raja@2024.07.0     umpire@2024.07.0
cabana@0.7.0       flecsi@2.3.2     kokkos@4.5.01     magma@2.9.0         slate@2024.10.29   umpire@2024.07.0
caliper@2.12.1     flux-core@0.67.0  kokkos@4.6.01     mfem@4.7.0          slepc@3.22.2       umpire@2024.07.0
camp@2024.07.0     ginkgo@1.9.0     kokkos@4.6.01     mgard@2023-12-09
==> 57 installed packages
Singularity> nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2025 NVIDIA Corporation
Built on Fri_Feb_21_20:23:50_PST_2025
Cuda compilation tools, release 12.8, V12.8.93
Build cuda_12.8.r12.8/compiler.35583870_0
Singularity> which huggingface-cli
/usr/local/bin/huggingface-cli
Singularity> which firefox
/usr/bin/firefox
Singularity> which codium
/usr/bin/codium
Singularity> which jupyter
/usr/local/bin/jupyter
Singularity> nvidia-smi
Sat Jun 7 22:02:16 2025
```

NVIDIA-SMI 570.124.06		Driver Version: 570.124.06		CUDA Version: 12.8	
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC
Fan	Temp	Pwr:Usage/Cap		Memory-Usage	GPU-Util Compute M.
					MIG M.
0	NVIDIA H100 PCIe	On	00000000:E1:00:0	Off	0



NVIDIA® BioNeMo™ Framework on E4S 25.06 CUDA x86_64

Activities **firefox-esr-esr128** Jun 8 05:12

sameer@illyad: ~/images/25.06

```
Singularity> ls
brca1 brca1_fasta_files compartmentalized-example nemo2_evo2_1b_8k zeroshot_brca1 zeroshot_brca1.ipynb
Singularity> jupyter notebook ./zeroshot_brca1.ipynb
[I 2025-06-07 22:06:28.260 ServerApp] jupyter_lsp | extension was successfully linked.
```

zeroshot_brca1 — Mozilla Firefox (on illyad)

zeroshot_brca1 x Brca1 Controls Homolo x evo2/notebooks/brca1 x +

localhost:8889/notebooks/zeroshot_brca1.ipynb

Jupyter zeroshot_brca1 Last Checkpoint: 3 days ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

Zero-shot prediction of BRCA1 variant effects with Evo 2

Deploy this tutorial on brev.dev: [Deploy Now](#)

Note - this notebook is a reproduction of The Arc Institute's same-titled notebook [here](#), using the [BioNeMo 2](#) implementation of Evo2.

Evo2 is a foundation AI model trained on 9.3 trillion DNA base pairs, predicting variant effects without prior task-specific training.

Without being explicitly trained on BRCA1 variants, we show Evo 2's ability to generalize across all life forms.

The human *BRCA1* gene encodes for a protein that repairs damaged DNA (Moynahan et al., 1999). Certain variants of this gene have been associated with an increased risk of breast and ovarian cancers (Miki et al., 1994). Using Evo 2, we can predict whether a particular single nucleotide variant (SNV) of the *BRCA1* gene is likely to be harmful to the protein's function, and thus potentially increase the risk of cancer for the patient with the genetic variant.

```
[1]: %%capture
import os

# Runs a subset of the model layers to test that the notebook runs in CI, but the output will be incorrect.
FAST_CI_MODE: bool = os.environ.get("FAST_CI_MODE", False)

[2]: import glob
import gzip
import json
import math
import os
from pathlib import Path

import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import torch
from Bio import SeqIO
from sklearn.metrics import auc, roc_auc_score, roc_curve
```

E4S 25.06 image for CUDA and x86_64 with VSCodium IDE

sameer@illyad:~/images/25.06

Singularity> codium zeroshot brca1.ipynb

zeroshot_brca1.ipynb - bionemo - VSCodium (on illyad)

File Edit Selection View Go Run Terminal Help

EXPLORER

OPEN EDITORS

zeroshot_brca1.ipynb M

zeroshot_brca1.ipynb > M • Zero-shot prediction of BRCA1 variant effects with Evo 2

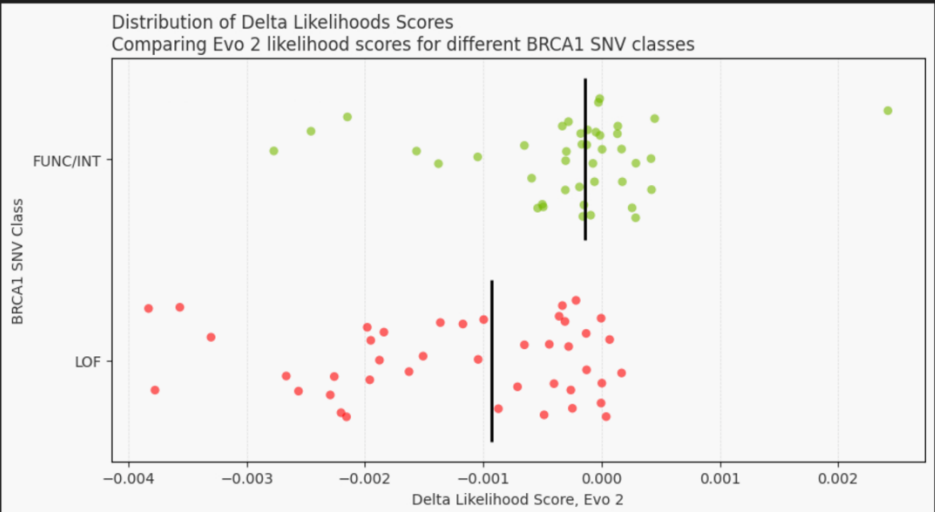
+ Code + Markdown ...

plot_strip_with_means(brca1_df, x_col="evo2_delta_score", class_col="class")

[17]

Python

Distribution of Delta Likelihood Scores
Comparing Evo 2 likelihood scores for different BRCA1 SNV classes



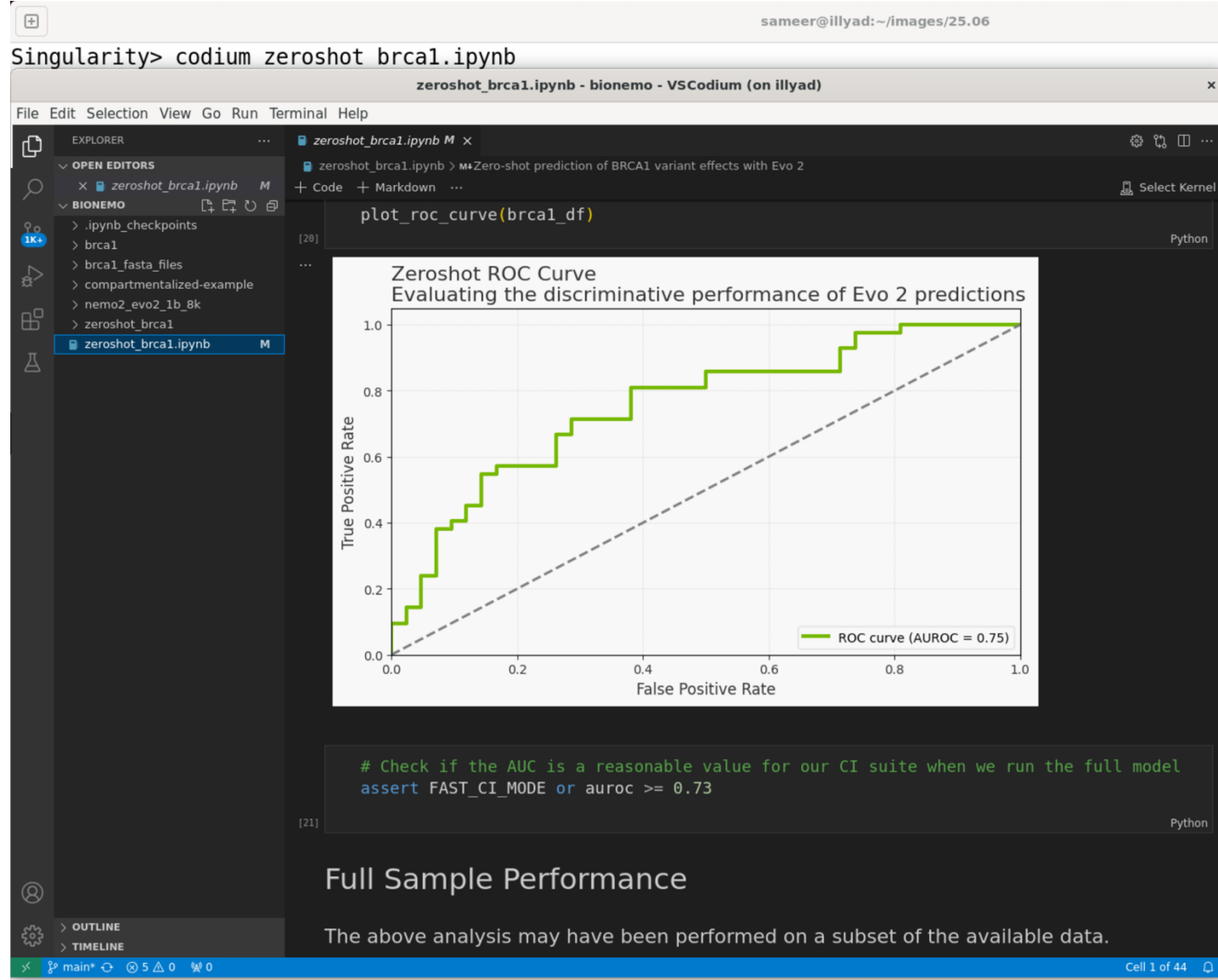
We can also calculate the area under the receiver operating characteristic curve (AUROC) of this zero-shot prediction method. Note that the results are nearly random unless you are on one of the following configurations:

- `--fp8` on an fp8 enabled GPU with either the 1b or 7b models. The 40b likely works as well.
- the 7b model uniquely seems to work well without `--fp8` so if you are on an older device, the 7b model should produce robust results. Change the `MODEL_SIZE` earlier in this tutorial and rerun for good results in that case.

Cell 1 of 44

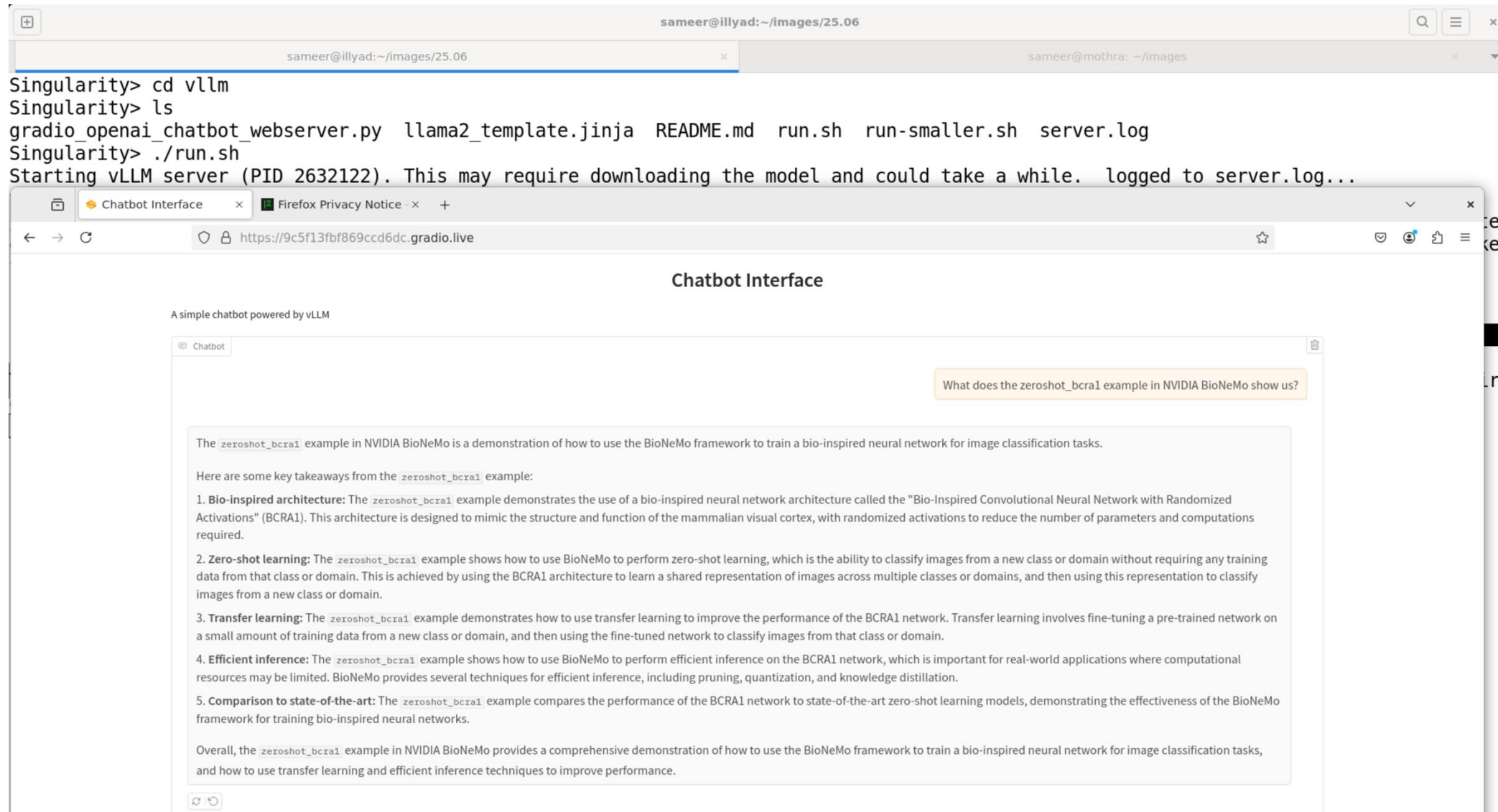
- NVIDIA H100 (cuda90) GPU on x86_64
- Jupyter Notebook in VSCodium IDE
- Running NVIDIA® BioNeMo™ Framework for biopharma workflows

E4S 25.06 image for CUDA and x86_64 with VSCodium IDE



- NVIDIA H100 (cuda90) GPU on x86_64
- Jupyter Notebook in VSCodium IDE
- Running NVIDIA® BioNeMo™ Framework for biopharma workflows

Creating a Chatbot using Vllm using E4S 25.06 image for x86_64



The screenshot shows a terminal window and a web browser. The terminal window, titled 'sameer@illyad: ~/images/25.06', displays the following commands and output:

```
Singularity> cd vllm
Singularity> ls
gradio_openai_chatbot_webserver.py  llama2_template.jinja  README.md  run.sh  run-smaller.sh  server.log
Singularity> ./run.sh
Starting vLLM server (PID 2632122). This may require downloading the model and could take a while. logged to server.log...
```

The web browser, titled 'sameer@mothra: ~/images', shows the 'Chatbot Interface' at the URL <https://9c5f13fbf869ccd6dc.gradio.live>. The interface is titled 'Chatbot Interface' and includes a subtitle 'A simple chatbot powered by vLLM'. It features a chat input field with the text 'What does the zeroshot_bcra1 example in NVIDIA BioNeMo show us?' and a response area containing the following text:

The `zeroshot_bcra1` example in NVIDIA BioNeMo is a demonstration of how to use the BioNeMo framework to train a bio-inspired neural network for image classification tasks.

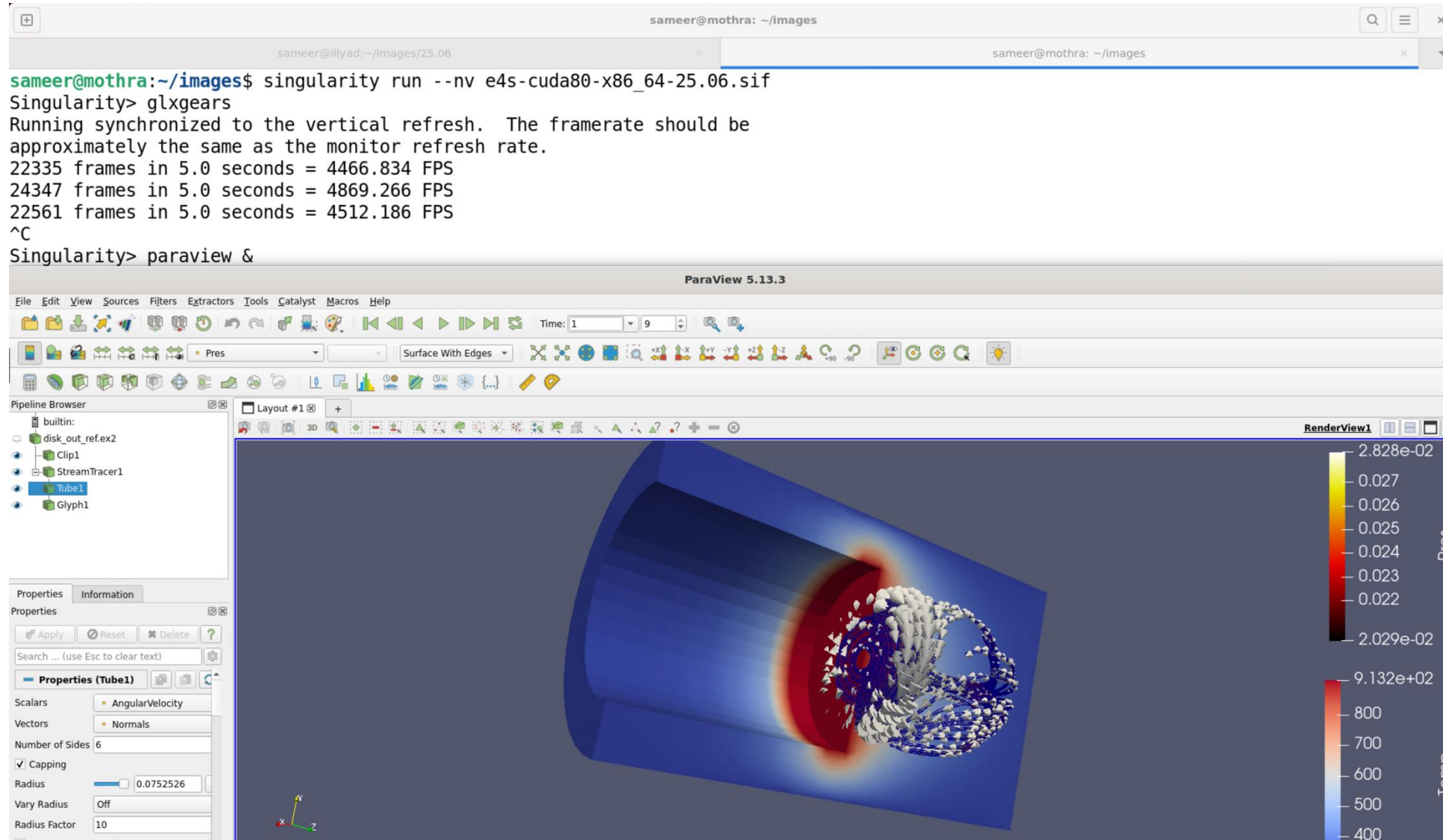
Here are some key takeaways from the `zeroshot_bcra1` example:

- Bio-inspired architecture:** The `zeroshot_bcra1` example demonstrates the use of a bio-inspired neural network architecture called the "Bio-Inspired Convolutional Neural Network with Randomized Activations" (BCRA1). This architecture is designed to mimic the structure and function of the mammalian visual cortex, with randomized activations to reduce the number of parameters and computations required.
- Zero-shot learning:** The `zeroshot_bcra1` example shows how to use BioNeMo to perform zero-shot learning, which is the ability to classify images from a new class or domain without requiring any training data from that class or domain. This is achieved by using the BCRA1 architecture to learn a shared representation of images across multiple classes or domains, and then using this representation to classify images from a new class or domain.
- Transfer learning:** The `zeroshot_bcra1` example demonstrates how to use transfer learning to improve the performance of the BCRA1 network. Transfer learning involves fine-tuning a pre-trained network on a small amount of training data from a new class or domain, and then using the fine-tuned network to classify images from that class or domain.
- Efficient inference:** The `zeroshot_bcra1` example shows how to use BioNeMo to perform efficient inference on the BCRA1 network, which is important for real-world applications where computational resources may be limited. BioNeMo provides several techniques for efficient inference, including pruning, quantization, and knowledge distillation.
- Comparison to state-of-the-art:** The `zeroshot_bcra1` example compares the performance of the BCRA1 network to state-of-the-art zero-shot learning models, demonstrating the effectiveness of the BioNeMo framework for training bio-inspired neural networks.

Overall, the `zeroshot_bcra1` example in NVIDIA BioNeMo provides a comprehensive demonstration of how to use the BioNeMo framework to train a bio-inspired neural network for image classification tasks, and how to use transfer learning and efficient inference techniques to improve performance.

- NVIDIA H100 (cuda90) GPU on x86_64
- Vllm chatbot running after huggingface-cli login
- Using local H100 GPU

GPU accelerated 3D graphics using E4S 25.06 image for x86_64

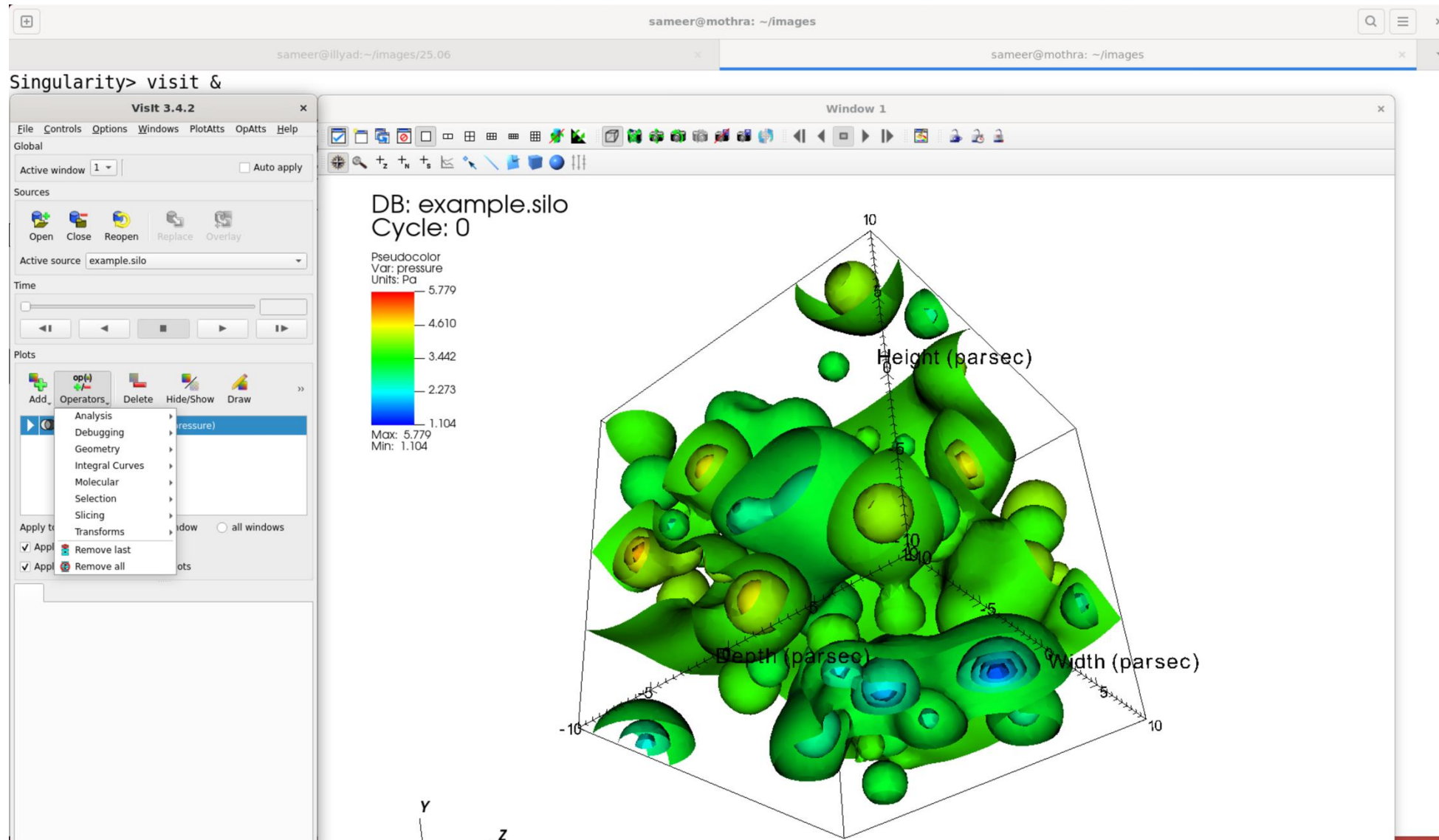


The screenshot displays a remote desktop environment. At the top, a terminal window shows the execution of a Singularity container command: `singularity run --nv e4s-cuda80-x86_64-25.06.sif`. Below this, the `glxgears` benchmark is run, displaying performance metrics: 22335 frames in 5.0 seconds = 4466.834 FPS, 24347 frames in 5.0 seconds = 4869.266 FPS, and 22561 frames in 5.0 seconds = 4512.186 FPS. The terminal also shows the execution of `paraview &`.

Below the terminal, the ParaView 5.13.3 application window is visible. It shows a 3D visualization of a fluid flow simulation. The visualization includes a color map for pressure (Pres) and temperature (Temp). The pressure scale ranges from $2.828e-02$ to $2.029e-02$, and the temperature scale ranges from $9.132e+02$ to 400 . The simulation shows a complex flow pattern within a cylindrical structure, with a color gradient indicating the distribution of pressure and temperature.

- Rendering on an NVIDIA A100 (cuda80) GPU on x86_64
- ParaView
- Using Adaptive Computing's Heidi/ODDC remote desktop

VisIt: GPU accelerated 3D graphics in E4S 25.06 image for x86_64



- Rendering on an NVIDIA A100 (cuda80) GPU on x86_64
- VisIt
- Using Adaptive Computing's Heidi/ODDC remote desktop

E4S 25.06 image for NVIDIA Grace-Hopper (cuda90, aarch64)

```
$ singularity run --nv e4s-cuda90-aarch64-25.06.sif
Singularity> uname -m
aarch64
Singularity> spack find +cuda
-- linux-ubuntu24.04-aarch64 / gcc@13.3.0 -----
adios2@2.10.2      camp@2024.07.0    ginkgo@1.9.0      kokkos@4.6.01     libpressio@0.99.4  petsc@3.22.4      tasmanian@8.1     zfp@1.0.0
amrex@25.03        chai@2024.07.0    gromacs@2024.4     kokkos@4.6.01     magma@2.9.0        raja@2024.07.0    tau@2.34.1
arborx@1.5         chapel@2.4.0      heffte@2.4.1      kokkos@4.6.01     mfem@4.7.0         raja@2024.07.0    trilinos@16.1.0
axom@0.10.1        cp2k@2025.1       hpctoolkit@2024.01.1 kokkos-kernels@4.6.01 mgard@2023-12-09  slate@2024.10.29  umpire@2024.07.0
blaspp@2024.10.26  cusz@0.14.0       hpx@1.10.0         lammps@20240829.1  nvcomp@2.2.0       slepc@3.22.2      umpire@2024.07.0
cabana@0.7.0       fftx@1.2.0        hwloc@2.11.1       lapackpp@2024.10.26 omega-h@10.8.6-scorec strumpack@8.0.0    umpire@2024.07.0
caliper@2.12.1     flecsi@2.3.2      hypre@2.32.0        legion@24.12.0     parsec@3.0.2209    sundials@7.2.1    upcxx@2023.9.0
camp@2024.07.0     flux-core@0.67.0  kokkos@4.5.01      libceed@0.12.0    petsc@3.22.4       superlu-dist@9.1.0 vtk-m@2.2.0
==> 57 installed packages
Singularity> nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2025 NVIDIA Corporation
Built on Fri_Feb_21_20:26:18_PST_2025
Cuda compilation tools, release 12.8, V12.8.93
Build cuda_12.8.r12.8/compiler.35583870_0
Singularity>
Singularity> nvidia-smi
Sat Jun 7 22:35:36 2025
```

NVIDIA-SMI 570.124.06				Driver Version: 570.124.06				CUDA Version: 12.8			
GPU	Name	Perf	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr.	ECC			
Fan	Temp		Pwr:Usage/Cap		Memory-Usage	GPU-Util	Compute	M.			
							MIG	M.			
0	NVIDIA GH200 480GB		On	000000009:01:00.0	Off			0			
N/A	27C	P0	122W / 700W	89270MiB / 97871MiB		1%	Default	Disabled			



E4S 25.06 image for x86_64 for AMD MI300A GPU (ROCm 6.3.3)

```
$ singularity run e4s-rocm942-x86_64-25.06.sif
Singularity> uname -m
x86_64
Singularity> lscpu | grep AMD
Vendor ID: AuthenticAMD
Model name: AMD Instinct MI300A Accelerator
Virtualization: AMD-V
Singularity> ls /opt
amdgpu demo rocm rocm-6.3.3
Singularity> spack find +rocm
-- linux-ubuntu22.04-x86_64_v3 / gcc@11.4.0 -----
amrex@25.03      chai@2024.07.0    hip@6.3.2        hpctoolkit@2024.01.1  kokkos@4.6.01    petsc@3.22.4      superlu-dist@9.1.0  vtk-m@2.2.0
arborx@1.7       chapel@2.4.0      hipblas@6.3.2    hpx@1.10.0          lapackpp@2024.10.26  raja@2024.07.0    tasmanian@8.1
blaspp@2024.10.26  fftx@1.2.0        hipfft@6.3.2     hypre@2.32.0         legion@24.12.0      slate@2024.10.29  tau@2.34.1
cabana@0.7.0      gasnet@2025.2.0-snapshot  hiprand@6.3.2    kokkos@3.7.01        libceed@0.12.0      slepc@3.22.2      trilinos@16.1.0
caliper@2.12.1    ginkgo@1.9.0      hipsolver@6.3.2  kokkos@4.5.01        mfem@4.7.0          strumpack@8.0.0   umpire@2024.07.0
camp@2024.07.0    heffte@2.4.1      hipsparse@6.3.2  kokkos@4.6.01        paraview@5.13.2     sundials@7.2.1    upcxx@2023.9.0
==> 43 installed packages
Singularity> which huggingface-cli
/usr/local/bin/huggingface-cli
Singularity> which codium
/usr/bin/codium
Singularity> which jupyter
/usr/local/bin/jupyter
Singularity> python
Python 3.10.12 (main, Feb 4 2025, 14:57:36) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> torch.cuda.get_arch_list()
['gfx900', 'gfx906', 'gfx908', 'gfx90a', 'gfx942', 'gfx1030', 'gfx1100', 'gfx1101', 'gfx1102', 'gfx1200', 'gfx1201']
>>> torch.__version__
'2.7.1+rocm6.3'
>>> import vllm
INFO 06-07 22:39:54 [__init__.py:248] Automatically detected platform rocm.
>>> █
```

E4S 25.06 image for CPU x86_64 without GPU runtimes (GCC)

```
$ singularity run e4s-cpu-x86_64-25.06.sif
```

```
Singularity> uname -m
```

```
x86_64
```

```
Singularity> spack find -x
```

```
-- linux-ubuntu22.04-x86_64_v3 / gcc@11.4.0 -----
```

adios@1.13.1	dyninst@13.0.0	hdf5-vol-async@1.7	mfem@4.7.0	petsc@3.22.4	sundials@7.2.1
alquimia@1.1.0	e4s-alc@1.0.2	hdf5-vol-cache@v1.1	mgard@2023-12-09	phist@1.12.1	superlu@7.0.0
aml@0.2.1	e4s-cl@1.0.4	hdf5-vol-log@1.4.0	mpark-variant@1.4.0	plasma@24.8.7	superlu-dist@9.1.0
amrex@25.03	exago@1.6.0	heffte@2.4.1	mpifileutils@0.12	plumed@2.9.2	swig@4.0.2-fortran
arborx@1.7	exaworks@0.1.0	hpctoolkit@2024.01.1	nccmp@1.9.1.0	precice@3.1.2	sz3@3.2.0
argobots@1.2	fftx@1.2.0	hpx@1.10.0	nco@5.3.2	pruners-ninja@1.0.1	tasmanian@8.1
axom@0.10.1	flecsi@2.3.2	hypre@2.32.0	nek5000@19.0	pumi@2.2.9	tau@2.34.1
bolt@2.0	flit@2.1.0	kokkos@4.6.01	nekbone@17.0	py-h5py@3.12.1	trilinos@16.1.0
boost@1.87.0	flux-core@0.67.0	kokkos-kernels@4.6.01	netcdf-fortran@4.6.1	py-jupyterhub@1.4.1	turbine@1.3.0
bricks@2023.08.25	fortrilinos@2.3.0	laghos@3.1	netlib-scalapack@2.2.2	py-libensemble@1.4.3	umap@2.1.1
butterflypack@3.2.0	fpm@0.10.0	lammps@20240829.1	nrm@0.1.0	py-petsc4py@3.22.4	umpire@2024.07.0
cabana@0.7.0	gasnet@2025.2.0-snapshot	lbann@0.104	nwchem@7.2.3	qthreads@1.18	upcxx@2023.9.0
caliper@2.12.1	ginkgo@1.9.0	legion@24.12.0	omega-h@10.8.6-scorec	quantum-espresso@7.4.1	variorum@0.8.0
chai@2024.07.0	globalarrays@5.8.2	libceed@0.12.0	openfoam@2412	raja@2024.07.0	wannier90@3.1.0
chapel@2.4.0	glvis@4.2	libnrm@0.1.0	openmpi@5.0.6	rempi@1.1.0	warpx@25.03
charliecloud@0.38	gmp@6.3.0	libquo@1.4	openpmd-api@0.16.1	scr@3.1.0	wps@4.5
conduit@0.9.3	gotcha@1.0.7	libunwind@1.8.1	papi@7.1.0	slate@2024.10.29	wrf@4.6.1
cp2k@2025.1	gptune@4.0.0	loki@0.1.7	papyrus@1.0.2	slepc@3.22.2	xyce@7.8.0
datatransferkit@3.1.1	gromacs@2024.4	mercury@2.4.0	parsec@3.0.2209	stc@0.9.0	
dealii@9.6.2	h5bench@1.4	metall@0.30	pdt@3.25.2	strumpack@8.0.0	

```
==> 118 installed packages
```

```
Singularity>
```

E4S 25.06 image for Intel GPUs with Intel compilers (x86_64)

```
$ singularity run e4s-oneapi-x86_64-25.06.sif
Singularity> uname -m
x86_64
Singularity> clinfo -l
Platform #0: Intel(R) OpenCL
`-- Device #0: Intel(R) Xeon(R) Silver 4410T
Platform #1: Intel(R) OpenCL Graphics
`-- Device #0: Intel(R) Data Center GPU Max 1100
Singularity> spack find -x
-- linux-ubuntu22.04-x86_64_v3 / oneapi@2025.1.0 -----
adios@1.13.1      darshan-runtime@3.4.6  hdf5-vol-cache@v1.1  mgard@2023-12-09      precice@3.1.2        sz@2.1.12.5
adios2@2.10.2    darshan-util@3.4.6    hdf5-vol-log@1.4.0  mpark-variant@1.4.0  pruners-ninja@1.0.1  sz3@3.2.0
alquimia@1.1.0   datatransferkit@3.1.1 heffte@2.4.1         mpifileutils@0.12     pumi@2.2.9           tasmanian@8.1
aml@0.2.1        e4s-alc@1.0.2         heffte@2.4.1         nccmp@1.9.1.0        py-cinemasci@1.7.0   tau@2.34.1
aml@0.2.1        e4s-cl@1.0.4         hpctoolkit@2024.01.1 nco@5.3.2            py-h5py@3.12.1       tau@2.34.1
amrex@25.03      exaworks@0.1.0       hpx@1.10.0          nekbone@17.0         py-jupyterhub@1.4.1  trilinos@16.1.0
amrex@25.03      faodel@1.2108.1      hypre@2.32.0        netcdf-fortran@4.6.1 py-libensemble@1.4.3  turbine@1.3.0
arborx@1.7       flecsi@2.3.2         kokkos@4.6.01       netlib-scalapack@2.2.2 py-petsc4py@3.22.4   umap@2.1.1
arborx@1.7       flit@2.1.0          kokkos@4.6.01       nrm@0.1.0           qthreads@1.18        umpire@2024.07.0
argobots@1.2     flux-core@0.67.0     kokkos-kernels@4.6.01 nwchem@7.2.3         quantum-espresso@7.4.1 unifyfs@2.0
ascent@0.9.3     forttrilinos@2.3.0   laghos@3.1          omega-h@10.8.6-scorec raja@2024.07.0        upcxx@2023.9.0
axom@0.10.1      gasnet@2025.2.0-snapshot lammps@20240829.1    openfoam@2412        rempi@1.1.0          upcxx@2023.9.0
bolt@2.0         ginkgo@1.9.0         legion@24.12.0      openmpi@5.0.6        scr@3.1.0            variorum@0.8.0
boost@1.87.0     ginkgo@1.9.0         libcatayst@2.0.0    openpmd-api@0.16.1  slate@2024.10.29     veloc@1.7
bricks@2023.08.25 globalarrays@5.8.2   libceed@0.12.0      papi@7.1.0           slepc@3.22.2         vtk-m@2.2.0
butterflypack@3.2.0 gmp@6.3.0          libnrm@0.1.0        papyrus@1.0.2        stc@0.9.0            wannier90@3.1.0
cabana@0.7.0     gotcha@1.0.7        libquo@1.4          parallel-netcdf@1.14.0 parsec@3.0.2209      sundials@7.2.1      xyce@7.8.0
cabana@0.7.0     gptune@4.0.0        libunwind@1.8.1     pdt@3.25.2          petsc@3.22.4         sundials@7.2.1      zfp@1.0.0
caliper@2.12.1   gromacs@2024.4      loki@0.1.7         phist@1.12.1         phist@1.12.1         superlu@7.0.0       superlu-dist@9.1.0
chai@2024.07.0   hdf5@1.14.5         metall@0.30         plumed@2.9.2        plumed@2.9.2        swig@4.0.2-fortran
conduit@0.9.3    hdf5-vol-async@1.7  mfem@4.7.0
==> 129 installed packages
Singularity> which ifx
/opt/intel/oneapi/compiler/2025.1/bin/ifx
```

E4S 25.06 image for Intel® oneAPI built with Intel MPI and compilers

```
Singularity> spack find -dl -v amrex +sycl
-- linux-ubuntu22.04-x86_64_v3 / oneapi@2025.1.0 -----
vibmei2 amrex@25.03~amrdata~ascent~catalyst~conduit~cuda+eb+fft~fortran~hdf5~hypr~ipo+linear_solvers+mpi~openmp~particles~petsc~pic~plotfile_tools~rocm~sha
red~sundials+sycl~tiny_profile build_system=cmake build_type=Release dimensions=1,2,3 generator=make precision=double
pkhuomn      cmake@3.31.6~doc+ncurses+ownlibs~qtgui build_system=generic build_type=Release
2kk4mv5      curl@8.11.1~gssapi~ldap~libidn2~librtmp~libssh~libssh2+nghttp2 build_system=autotools libs=shared,static tls=openssl
vizwdui      nghttp2@1.65.0 build_system=autotools
izwrsao      diffutils@3.10 build_system=autotools
y2xtcez      libiconv@1.17 build_system=autotools libs=shared,static
avu5xay      openssl@3.4.1~docs+shared build_system=generic certs=mozilla
v4unosi      ca-certificates-mozilla@2025-02-25 build_system=generic
27y67gw      perl@5.40.0+cpanm+opcode+open+shared+threads build_system=generic
mie6fpa      berkeley-db@18.1.40+cxx~docs+stl build_system=autotools patches=26090f4,b231fcc
eeheopa      bzip2@1.0.8~debug~pic+shared build_system=generic
cpcojya      gdbm@1.23 build_system=autotools
kud5giw      readline@8.2 build_system=autotools patches=1ea4349,24f587b,3d9885e,5911a5b,622ba38,6c8adf8,758e2ec,79572ee,a177edc,bbf97f1,
c7b45ff,e0013d9,e065038
rv3c6cm      ncurses@6.5~symlinks+termlib abi=none build_system=autotools patches=7a351bc
jzthfaz      zlib-ng@2.2.3+compat+new_strategies+opt+pic+shared build_system=autotools
7qebqtu      glibc@2.35 build_system=autotools
nx5zggc      gmake@4.4.1~guile build_system=generic
lzycjnj      intel-oneapi-mkl@2025.1.0~cluster+envmods~gfortran~ilp64+shared build_system=generic mpi_family=none threads=none
nxlzht7      intel-oneapi-mpi@2021.15.0~classic~names+envmods~external-libfabric~generic~names~ilp64 build_system=generic
3obsopm      intel-oneapi-runtime@2025.1.0 build_system=generic
ncwe67l      gcc-runtime@11.4.0 build_system=generic
2ianxzp      pkgconf@2.3.0 build_system=autotools
```

==> 1 installed package
Singularity>

Intel oneAPI based packages in E4S 25.06 image for x86_64

```
$ singularity run e4s-oneapi-x86_64-25.06.sif
Singularity> module avail
```

```
----- /spack/share/spack/lmod/linux-ubuntu22.04-x86_64/intel-oneapi-mpi/2021.15.0-nxlzht7/Core -----
adios/1.13.1          ginkgo/1.9.0-sycl-openmp (D)  nccmp/1.9.1.0          rempi/1.1.0
adios2/2.10.2         globalarrays/5.8.2          nco/5.3.2-openmp       scr/3.1.0
alquimia/1.1.0        gptune/4.0.0                nekbone/17.0           slate/2024.10.29-openmp
amrex/25.03-sycl      gromacs/2024.4-openmp       netcdf-fortran/4.6.1   slepc/3.22.2
amrex/25.03           hdf5bench/1.4               netlib-scalapack/2.2.2 stc/0.9.0
arborx/1.7-sycl      hdf5-vol-async/1.7          nwchem/7.2.3           strumpack/8.0.0-openmp
arborx/1.7           hdf5-vol-cache/v1.1        omega-h/10.8.6-scorec sundials/7.2.1-sycl
ascent/0.9.3-openmp  hdf5-vol-log/1.4.0         openfoam/2412          sundials/7.2.1 (D)
axom/0.10.1-openmp   hdf5/1.14.5                openpmd-api/0.16.1     superlu-dist/9.1.0
bricks/2023.08.25    heffte/2.4.1-sycl          papyrus/1.0.2          tasmanian/8.1
butterflypack/3.2.0-openmp heffte/2.4.1 (D)         parallel-netcdf/1.14.0 tau/2.34.1-level-zero
cabana/0.7.0-sycl    hpctoolkit/2024.01.1       parsec/3.0.2209        tau/2.34.1 (D)
cabana/0.7.0         hpx/1.10.0                 petsc/3.22.4           trilinos/16.1.0
caliper/2.12.1       hypre/2.32.0               phist/1.12.1-openmp   turbine/1.3.0
chai/2024.07.0       laghos/3.1                 plumed/2.9.2           umpire/2024.07.0
conduit/0.9.3        lammmps/20240829.1-openmp  precice/3.1.2          unifyfs/2.0
darshan-runtime/3.4.6 libcatayst/2.0.0           pruners-ninja/1.0.1   upcxx/2023.9.0-level-zero
datatransferkit/3.1.1 libnm/0.1.0                pumi/2.2.9            upcxx/2023.9.0 (D)
exaworks/0.1.0       libquo/1.4                 py-cinemasci/1.7.0    veloc/1.7
faodel/1.2108.1      mercury/2.4.0              py-h5py/3.12.1        vtk-m/2.2.0
flecsi/2.3.2         metall/0.30                py-libensemble/1.4.3  wannier90/3.1.0
fortrilinos/2.3.0    mfem/4.7.0                 py-petsc4py/3.22.4    wrf/4.6.1
ginkgo/1.9.0-openmp  mpifileutils/0.12          quantum-espresso/7.4.1-openmp xyce/7.8.0

----- /spack/share/spack/lmod/linux-ubuntu22.04-x86_64/Core -----
aml/0.2.1-level-zero  e4s-cl/1.0.4                kokkos/4.6.01-openmp   nrm/0.1.0              swig/4.0.2-fortran
aml/0.2.1             flit/2.1.0                  kokkos/4.6.01-sycl-openmp (D) openmpi/5.0.6          sz/2.1.12.5
argobots/1.2          flux-core/0.67.0            legion/24.12.0          papi/7.1.0             sz3/3.2.0
bolt/2.0              gasnet/2025.2.0-snapshot    libceed/0.12.0          pdt/3.25.2             umap/2.1.1
boost/1.87.0          gmp/6.3.0                  libunwind/1.8.1        py-jupyterhub/1.4.1    variorum/0.8.0
charliecloud/0.38    gotcha/1.0.7                loki/0.1.7             qthreads/1.18          zfp/1.0.0
darshan-util/3.4.6    intel-oneapi-mpi/2021.15.0 (L) mgard/2023-12-09-openmp raja/2024.07.0
e4s-alc/1.0.2         kokkos-kernels/4.6.01-openmp mpark-variant/1.4.0    superlu/7.0.0

----- /opt/intel/oneapi/modulefiles -----
compiler-intel-llvm/latest      compiler/latest      dev-utilities/latest      mkl/latest           tbb/latest
compiler-intel-llvm/2025.1.1 (D) compiler/2025.1.1 (D) dev-utilities/2025.1.0 (D) mkl/2025.1 (D)       tbb/2022.1 (D)
compiler-rt/latest             debugger/latest      dpl/latest                mpi/latest           umf/latest
compiler-rt/2025.1.1 (D)       debugger/2025.1.0 (D) dpl/2022.8                mpi/2021.15 (L,D)    umf/0.10.0 (D)
```

Where:

L: Module is loaded

D: Default Module



<https://e4s.io>

PyTorch built for Intel XPU in E4S 25.06 Intel® oneAPI image

```
$ singularity run e4s-oneapi-x86_64-25.06.sif
Singularity> uname -m
x86_64
Singularity> python
Python 3.10.12 (main, Feb  4 2025, 14:57:36) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> import intel_extension_for_pytorch as ipex
[W607 22:52:26.365045146 OperatorEntry.cpp:154] Warning: Warning only once for all operators, other operators may also be overridden.
  Overriding a previously registered kernel for the same operator and the same dispatch key
  operator: aten::geometric_(Tensor(a!) self, float p, *, Generator? generator=None) -> Tensor(a!)
    registered at /pytorch/build/aten/src/ATen/RegisterSchema.cpp:6
  dispatch key: XPU
  previous kernel: registered at /pytorch/aten/src/ATen/VmapModeRegistrations.cpp:37
  new kernel: registered at /build/intel-pytorch-extension/build/Release/csrc/gpu/csrc/gpu/xpu/ATen/RegisterXPU_0.cpp:186 (function operator())
>>> torch.xpu.is_available()
True
>>> torch.__version__
'2.7.0+xpu'
>>>
```

E4S 25.06 image for ppc64le with NVIDIA GPUs (IBM Power 9)

```
$ singularity run --nv e4s-cuda70-ppc64le-25.06.sif
```

```
Singularity> uname -m
```

```
ppc64le
```

```
Singularity> spack find -x
```

```
-- linux-ubuntu20.04-ppc64le / gcc@9.4.0 -----
```

adios@1.13.1	darshan-runtime@3.4.6	hdf5@1.14.5	mercury@2.4.0	pdtd@3.25.2	sundials@7.2.1
adios2@2.10.2	darshan-util@3.4.6	hdf5-vol-async@1.7	metall@0.30	petsc@3.22.4	superlu@7.0.0
alquimia@1.1.0	datatransferkit@3.1.1	hdf5-vol-cache@v1.1	mfem@4.7.0	petsc@3.22.4	superlu-dist@9.1.0
aml@0.2.1	dyninst@13.0.0	hdf5-vol-log@1.4.0	mfem@4.7.0	phist@1.12.1	superlu-dist@9.1.0
amrex@25.03	e4s-alc@1.0.2	heffte@2.4.1	mgard@2023-12-09	plasma@24.8.7	swig@4.0.2-fortran
amrex@25.03	e4s-cl@1.0.4	heffte@2.4.1	mgard@2023-12-09	plumed@2.9.2	sz@2.1.12.5
arborx@1.5	exago@1.6.0	hpctoolkit@2024.01.1	mpark-variant@1.4.0	precice@3.1.2	sz3@3.2.0
arborx@1.7	exago@1.6.0	hpctoolkit@2024.01.1	mpifileutils@0.12	pruners-ninja@1.0.1	tasmanian@8.1
argobots@1.2	exaworks@0.1.0	hpx@1.10.0	nccmp@1.9.1.0	pumi@2.2.9	tasmanian@8.1
ascent@0.9.3	faodel@1.2108.1	hpx@1.10.0	nco@5.3.2	py-cinemasci@1.7.0	tau@2.34.1
axom@0.10.1	fftx@1.2.0	hypre@2.32.0	nek5000@19.0	py-h5py@3.12.1	tau@2.34.1
axom@0.10.1	fftx@1.2.0	hypre@2.32.0	nekbone@17.0	py-jupyterhub@1.4.1	trilinos@16.1.0
bolt@2.0	flit@2.1.0	kokkos@4.6.01	netcdf-fortran@4.6.1	py-libensemble@1.4.3	turbine@1.3.0
boost@1.87.0	flux-core@0.67.0	kokkos@4.6.01	netlib-scalapack@2.2.2	py-petsc4py@3.22.4	umap@2.1.1
bricks@2023.08.25	flux-core@0.67.0	kokkos-kernels@4.6.01	nrm@0.1.0	qthreads@1.18	umpire@2024.07.0
bricks@2023.08.25	fortrilinos@2.3.0	kokkos-kernels@4.6.01	nwchem@7.2.3	quantum-espresso@7.4.1	umpire@2024.07.0
butterflypack@3.2.0	fpm@0.10.0	laghos@3.1	omega-h@10.8.6-scorec	raja@2024.07.0	unifyfs@2.0
cabana@0.7.0	gasnet@2025.2.0-snapshot	lammps@20240829.1	omega-h@10.8.6-scorec	raja@2024.07.0	upcxx@2023.9.0
cabana@0.7.0	ginkgo@1.9.0	lammps@20240829.1	openfoam@2412	rempi@1.1.0	upcxx@2023.9.0
caliper@2.12.1	ginkgo@1.9.0	lbann@0.104	openmpi@5.0.6	scr@3.1.0	variorum@0.8.0
caliper@2.12.1	globalarrays@5.8.2	legion@24.12.0	openpmd-api@0.16.1	slate@2024.10.29	veloc@1.7
chai@2024.07.0	glvis@4.2	libcatalyst@2.0.0	papi@7.1.0	slate@2024.10.29	visit@3.4.1
chai@2024.07.0	gmp@6.3.0	libnrm@0.1.0	papi@7.1.0	slepc@3.22.2	vtk-m@2.2.0
chapel@2.4.0	gotcha@1.0.7	libpressio@0.99.4	papyrus@1.0.2	slepc@3.22.2	wannier90@3.1.0
charliecloud@0.38	gptune@4.0.0	libquo@1.4	parallel-netcdf@1.14.0	stc@0.9.0	wps@4.5
conduit@0.9.3	gromacs@2024.4	libunwind@1.7.2	paraview@5.13.2	strumpack@8.0.0	wrf@4.6.1
cp2k@2025.1	gromacs@2024.4	loki@0.1.7	parsec@3.0.2209	strumpack@8.0.0	xyce@7.8.0
cp2k@2025.1	h5bench@1.4	magma@2.9.0	parsec@3.0.2209	sundials@7.2.1	zfp@1.0.0

```
==> 168 installed packages
```

```
Singularity> █
```

```
$ singularity run --nv e4s-cuda70-ppc64le-25.06.sif
```

```
Singularity> spack find +cuda
```

```
-- linux-ubuntu20.04-ppc64le / gcc@9.4.0 -----
```

amrex@25.03	camp@2024.07.0	gromacs@2024.4	kokkos@4.6.01	nvcomp@2.2.0	raja@2024.07.0	umpire@6.0.0
arborx@1.5	camp@2024.07.0	heffte@2.4.1	kokkos@4.6.01	omega-h@10.8.6-scorec	slate@2024.10.29	umpire@2024.07.0
axom@0.10.1	chai@2024.07.0	hiop@1.0.0	kokkos-kernels@4.6.01	papi@7.1.0	slepc@3.22.2	umpire@2024.07.0
blaspp@2024.10.26	cp2k@2025.1	hpctoolkit@2024.01.1	lammps@20240829.1	parsec@3.0.2209	strumpack@8.0.0	umpire@2024.07.0
bricks@2023.08.25	exago@1.6.0	hpx@1.10.0	lapackpp@2024.10.26	petsc@3.22.4	sundials@7.2.1	upcxx@2023.9.0
cabana@0.7.0	fftx@1.2.0	hwloc@2.11.1	magma@2.9.0	petsc@3.22.4	superlu-dist@9.1.0	
caliper@2.12.1	flux-core@0.67.0	hypre@2.32.0	mfem@4.7.0	raja@0.14.0	tasmanian@8.1	
camp@0.2.3	ginkgo@1.9.0	kokkos@4.6.01	mgard@2023-12-09	raja@2024.07.0	tau@2.34.1	

```
==> 53 installed packages
```

```
Singularity> which codium
```

```
/usr/local/codium/bin/codium
```

```
Singularity>
```

<https://e4s.io>



E4S: An HPC-AI Software Ecosystem for Science!

The screenshot displays a terminal window with the following content:

```
Singularity> spack find +cuda
-- linux-ubuntu22.04-x86_64_v3 / gcc@11.4.0 -----
adios2@2.10.2      camp@2024.07.0  flux-core@0.67.0  kokkos@4.5.01    libceed@0.12.0  petsc@3.22.4    tasmanian@8.1     zfp@1.0.0
amrex@25.03       camp@2024.07.0  ginkgo@1.9.0      kokkos@4.6.01    magma@2.9.0     raja@2024.07.0  tau@2.34.1
arborx@1.5        chai@2024.07.0  gromacs@2024.4     kokkos@4.6.01    mfem@4.7.0      raja@2024.07.0  trilinos@16.1.0
axom@0.10.1       chapel@2.4.0    heffte@2.4.1      kokkos@4.6.01    mgard@2023-12-09 slate@2024.10.29 umpire@2024.07.0
blaspp@2024.10.26 cp2k@2025.1     hpctoolkit@2024.01.1 kokkos-kernels@4.6.01 nvcomp@2.2.0    slepc@3.22.2     umpire@2024.07.0
bricks@2023.08.25 cusz@0.14.0     hpx@1.10.0        lammmps@20240829.1 papi@7.1.0      strumpack@8.0.0  umpire@2024.07.0
cabana@0.7.0      fftx@1.2.0      hwloc@2.11.1      lapackpp@2024.10.26 parsec@3.0.2209 sundials@7.2.1   upcxx@2023.9.0
caliper@2.12.1    flecsi@2.3.2    hypre@2.32.0      legion@24.12.0   petsc@3.22.4    superlu-dist@9.1.0 vtk-m@2.2.0

==> 57 installed packages
Singularity> paraprof demo.ppk &
```

Below the terminal, there are two windows:

- foo.py - nemo-speech_to_text - VSCode (on illyad)**: A code editor showing a Python script for converting .sph files to .wav and using the NeMo ASR model.
- TAU: ParaProf: 3D Visualizer: demo.ppk (on illyad)**: A 3D visualization of a data set, showing a color-coded surface plot with axes labeled 'seconds' and 'node'.

Acknowledgment

- *This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Next-Generation Scientific Software Technologies program, under contract numbers DE-AC02-AC05-00OR22725 and DOE SBIR DE-SC0022502.*



U.S. DEPARTMENT OF
ENERGY

Office of
Science

- <https://science.osti.gov/ascr>
- <https://pesoproject.org>
- <https://ascr-step.org>
- <https://hpsf.io>
- <https://www.energy.gov/technologytransitions/sbirsttr>

Thank you

<https://www.exascaleproject.org>

This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative.



Thank you to all collaborators in the ECP and broader computational science communities. The work discussed in this presentation represents creative contributions of many people who are passionately working toward next-generation computational science.

