

ParaTools Pro for E4S™: An HPC-AI ecosystem for science

13th Annual MVAPICH Users Group Meeting

11:15am – noon ET, Monday, August 18th, 2025
The Ohio State University, Columbus, OH



HPSF
HIGH PERFORMANCE
SOFTWARE FOUNDATION



Sameer Shende
Research Professor and Director,
Performance Research Laboratory, OACISS, University of Oregon
President and Director, ParaTools, Inc.
sameer@cs.uoregon.edu
https://oaciss.uoregon.edu/e4s/talks/E4S_MUG25.pdf



UNIVERSITY
OF OREGON

ParaTools



Motivation

- As our software gets more complex, it is getting harder and harder to install and operate our HPC-AI workflows on GPUs correctly in an integrated and interoperable software stack on commercial cloud platforms!
- What makes for a productive HPC-AI environment on the cloud platforms?

Extreme-scale Scientific Software Stack (E4S)

<https://e4s.io>



E4S: Extreme-scale Scientific Software Stack




<https://e4s.io>

About E4S

- E4S, an HPSF project, is an HPC-AI software **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 100+ 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 where available).
- Base images and full featured containers (with GPU support).
- Commercial support for E4S through ParaTools, Inc. for installation, maintaining an issue tracker, and application engagement.
- E4S for commercial clouds: MVAPICH and Adaptive's Heidi 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 address bar displays <https://e4s.io>. The main header features the title "E4S: An HPC-AI Software Ecosystem for Science" in large white text on a dark, abstract background. Below the title is a navigation menu 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 contains three paragraphs of text describing the E4S ecosystem, its support for scientific applications, and its origins in the US Exascale Computing Project.

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.

E4S Container Download from https://e4s.io

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

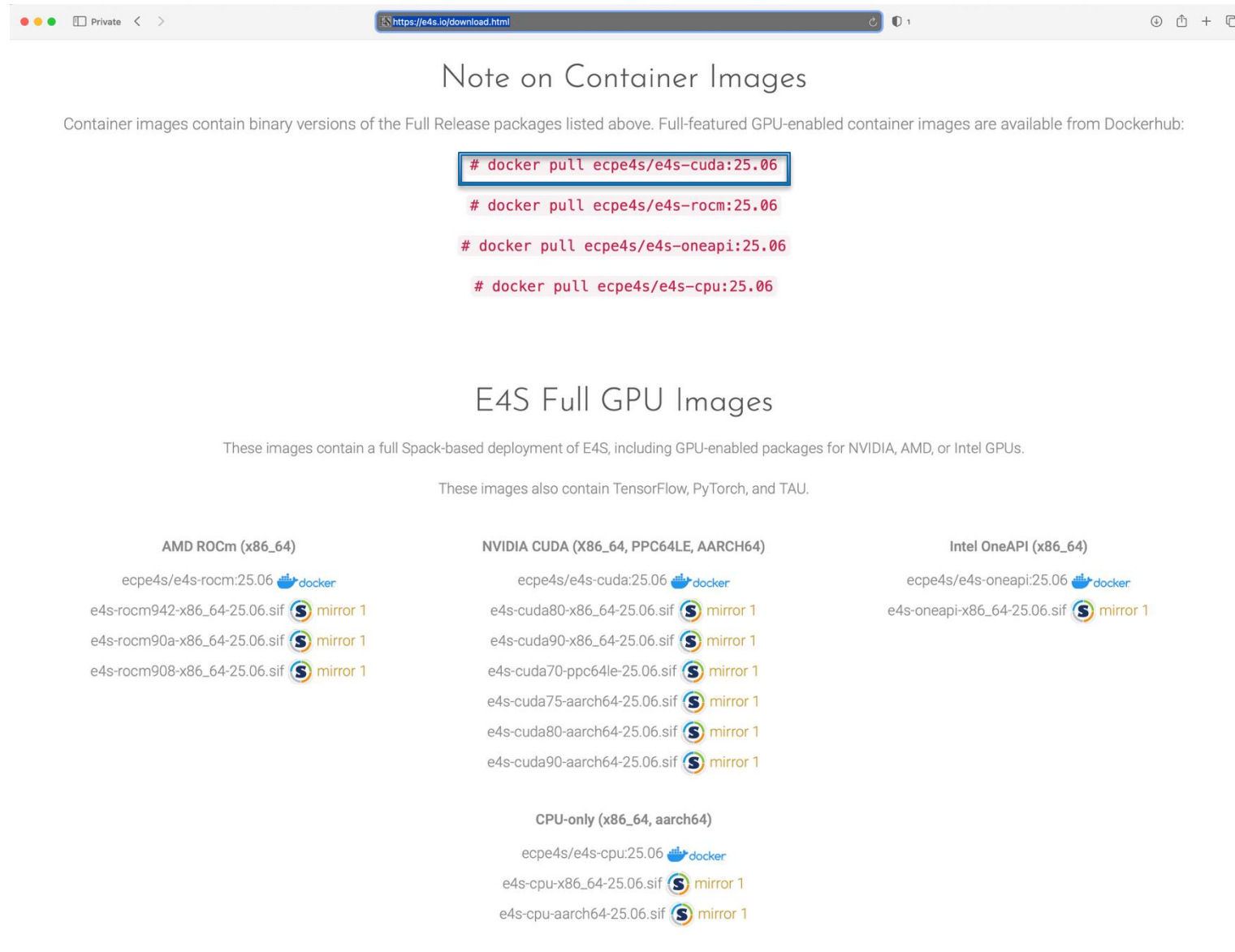
From source with Spack

[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.

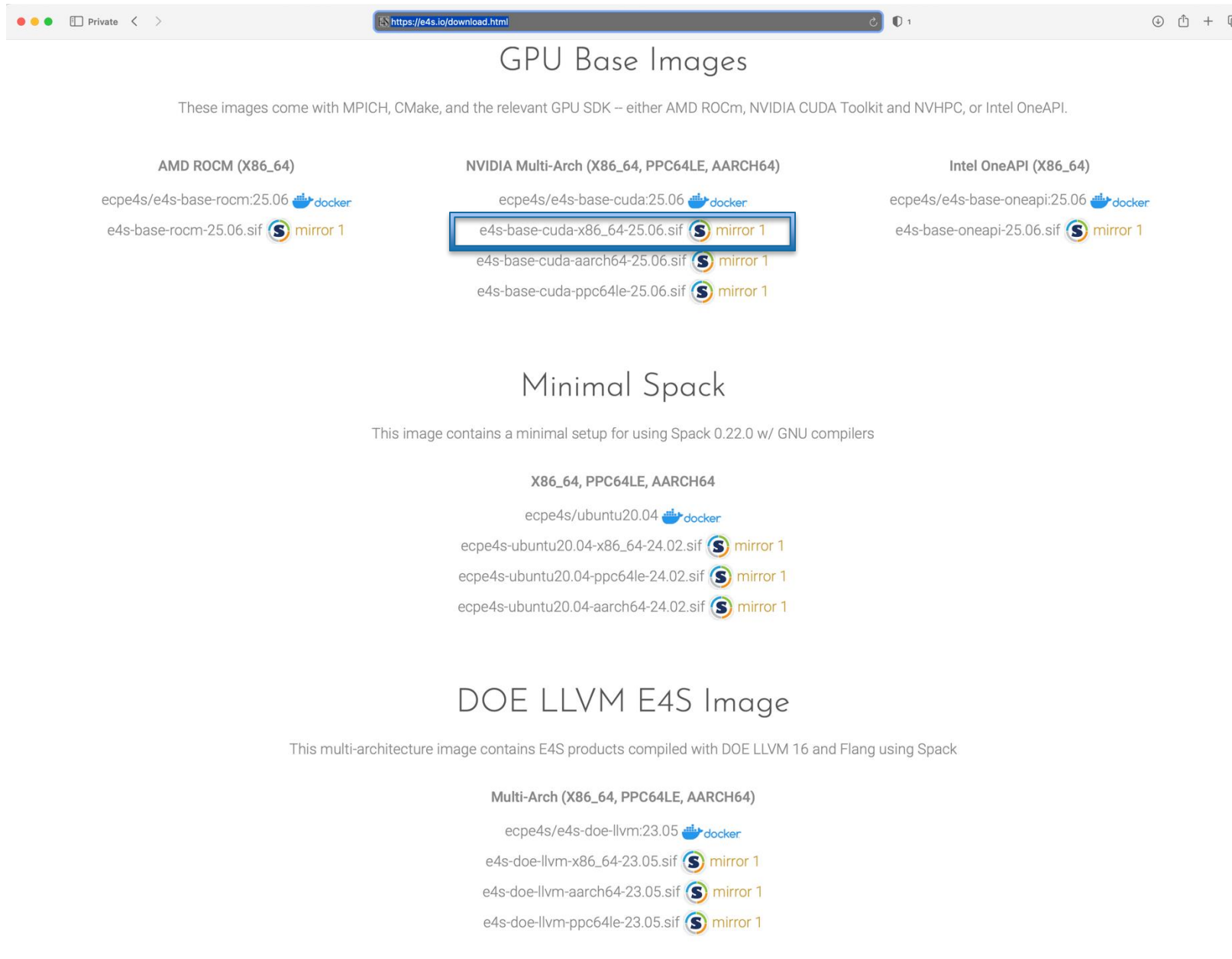
- Separate full featured Singularity images for 3 GPU architectures
- GPU full featured images for
 - x86_64 (Intel, AMD, NVIDIA)
 - ppc64le (NVIDIA)
 - aarch64 (NVIDIA)
- Full featured images available on Dockerhub
- 130+ products on 3 architectures

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











<https://e4s.io>





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 a description: "These images come with MPICH, CMake, and the relevant GPU SDK -- either AMD ROCm, NVIDIA CUDA Toolkit and NVHPC, or Intel OneAPI." Below this, there are three columns of container images:

- AMD ROCm (X86_64)**
 - ecpe4s/e4s-base-rocm:25.06 
 - e4s-base-rocm-25.06.sif  mirror 1
- NVIDIA Multi-Arch (X86_64, PPC64LE, AARCH64)**
 - ecpe4s/e4s-base-cuda:25.06 
 - e4s-base-cuda-x86_64-25.06.sif  mirror 1
 - e4s-base-cuda-aarch64-25.06.sif  mirror 1
 - e4s-base-cuda-ppc64le-25.06.sif  mirror 1
- Intel OneAPI (X86_64)**
 - ecpe4s/e4s-base-oneapi:25.06 
 - e4s-base-oneapi-25.06.sif  mirror 1

Below the GPU Base Images section, there is a section titled "Minimal Spack" with the description: "This image contains a minimal setup for using Spack 0.22.0 w/ GNU compilers". It lists the following images:

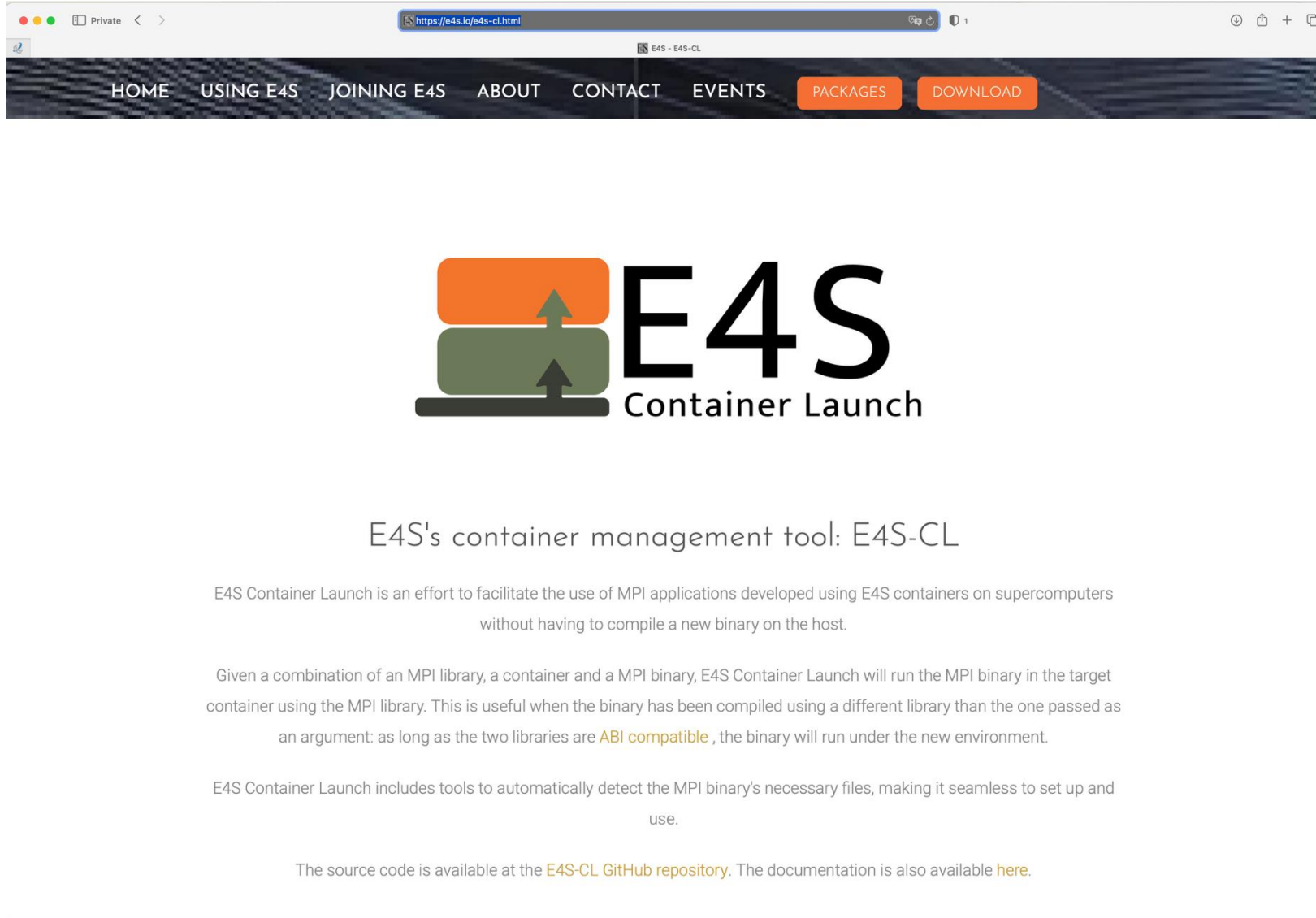
- X86_64, PPC64LE, AARCH64**
 - ecpe4s/ubuntu20.04 
 - ecpe4s-ubuntu20.04-x86_64-24.02.sif  mirror 1
 - ecpe4s-ubuntu20.04-ppc64le-24.02.sif  mirror 1
 - ecpe4s-ubuntu20.04-aarch64-24.02.sif  mirror 1

Below the Minimal Spack section, there is a section titled "DOE LLVM E4S Image" with the description: "This multi-architecture image contains E4S products compiled with DOE LLVM 16 and Flang using Spack". It lists the following images:


- Multi-Arch (X86_64, PPC64LE, AARCH64)**
 - ecpe4s/e4s-doe-llvm:23.05 
 - e4s-doe-llvm-x86_64-23.05.sif  mirror 1
 - e4s-doe-llvm-aarch64-23.05.sif  mirror 1
 - e4s-doe-llvm-ppc64le-23.05.sif  mirror 1

- Intel oneAPI
- AMD ROCm
- NVIDIA CUDA

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

A screenshot of a web browser displaying the E4S Container Launch website. The browser's address bar shows the URL "https://e4s.io/e4s-cl.html". The website has a dark header with navigation links: HOME, USING E4S, JOINING E4S, ABOUT, CONTACT, EVENTS, PACKAGES, and DOWNLOAD. The main content area features the E4S logo, which consists of two stacked rectangles (orange on top, green on bottom) with a green upward arrow between them, followed by the text "E4S Container Launch". Below the logo, the text "E4S's container management tool: E4S-CL" is displayed. Further down, there are three paragraphs of text explaining the tool's purpose and usage. The first paragraph states that E4S Container Launch facilitates the use of MPI applications developed using E4S containers on supercomputers without needing to compile a new binary on the host. The second paragraph explains that given a combination of an MPI library, a container, and a MPI binary, E4S Container Launch will run the MPI binary in the target container using the MPI library, which is useful when the binary has been compiled using a different library than the one passed as an argument, as long as the two libraries are ABI compatible. The third paragraph mentions that E4S Container Launch includes tools to automatically detect the MPI binary's necessary files, making it seamless to set up and use. At the bottom, it states that the source code is available at the E4S-CL GitHub repository and that the documentation is also available here.

HOME USING E4S JOINING E4S ABOUT CONTACT EVENTS PACKAGES DOWNLOAD



E4S Container Launch

E4S's container management tool: E4S-CL

E4S Container Launch is an effort to facilitate the use of MPI applications developed using E4S containers on supercomputers without having to compile a new binary on the host.

Given a combination of an MPI library, a container and a MPI binary, E4S Container Launch will run the MPI binary in the target container using the MPI library. This is useful when the binary has been compiled using a different library than the one passed as an argument: as long as the two libraries are **ABI compatible**, the binary will run under the new environment.

E4S Container Launch includes tools to automatically detect the MPI binary's necessary files, making it seamless to set up and use.

The source code is available at the [E4S-CL GitHub repository](#). The documentation is also available [here](#).

- 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

- 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.

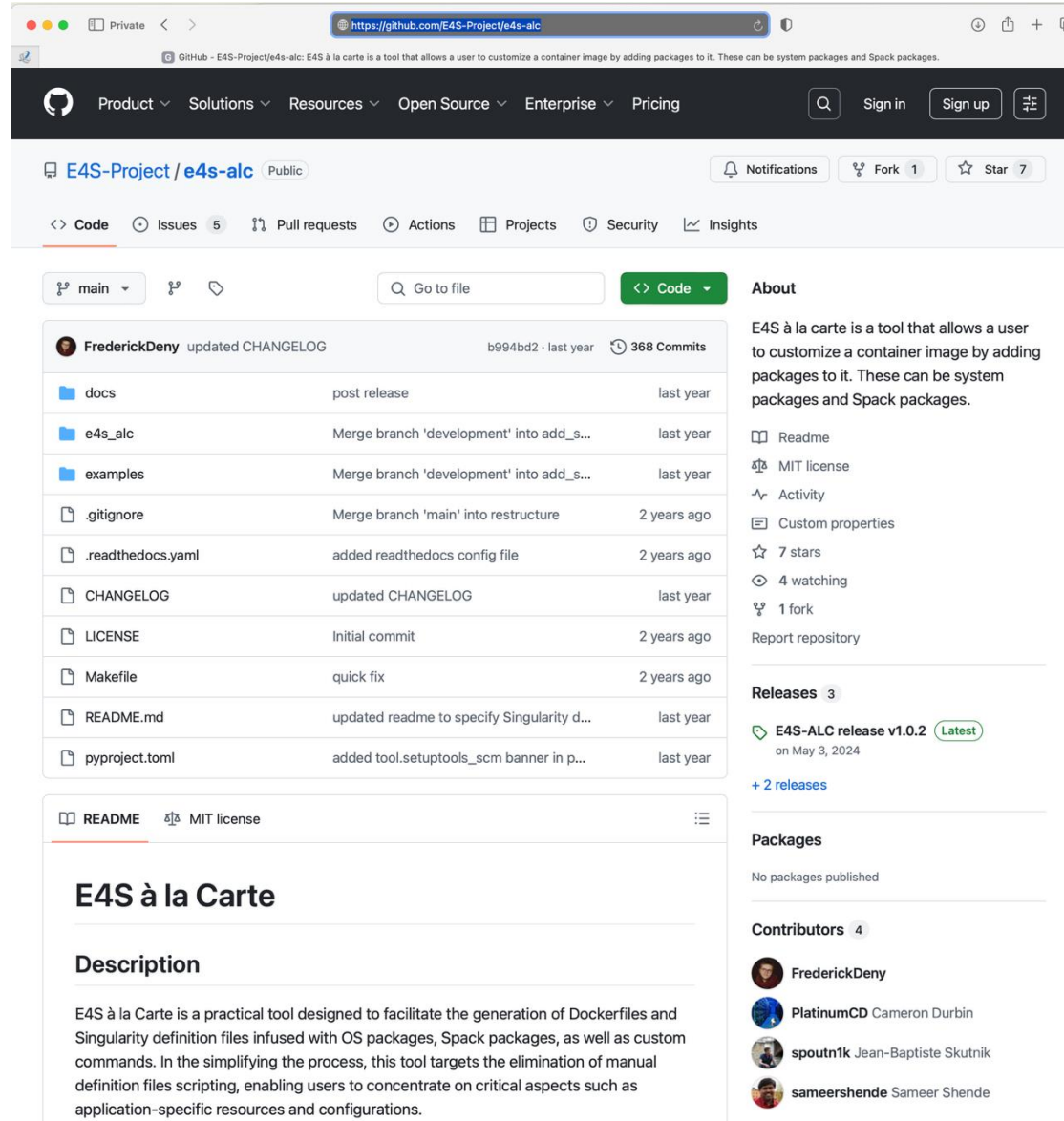
- e4s-cl init --backend [singularity|shifter|docker] --image <file> --source <startup_cmds.sh>
 - e4s-cl mpirun -np <N> <command>

- 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 Tools: E4S à la carte or e4s-alc: Customize container images



The screenshot displays the GitHub repository for `E4S-Project/e4s-alc`. The repository is public and has 7 stars and 1 fork. The main branch is selected. The repository structure includes folders like `docs`, `e4s_alc`, and `examples`, and files like `.gitignore`, `.readthedocs.yaml`, `CHANGELOG`, `LICENSE`, `Makefile`, `README.md`, and `pyproject.toml`. The README is titled "E4S à la Carte" and describes the tool as a practical tool for generating Dockerfiles and Singularity definition files.

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.

- 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!

Spack

- E4S uses the Spack package manager for software delivery
- Spack provides the ability to specify versions of software packages that are and are not interoperable.
- Spack is a build layer for not only E4S software, but also a large collection of software tools and libraries outside of ECP ST.
- Spack supports achieving and maintaining interoperability between ST software packages.
- <https://spack.io>

Spack is a flexible package manager for HPC

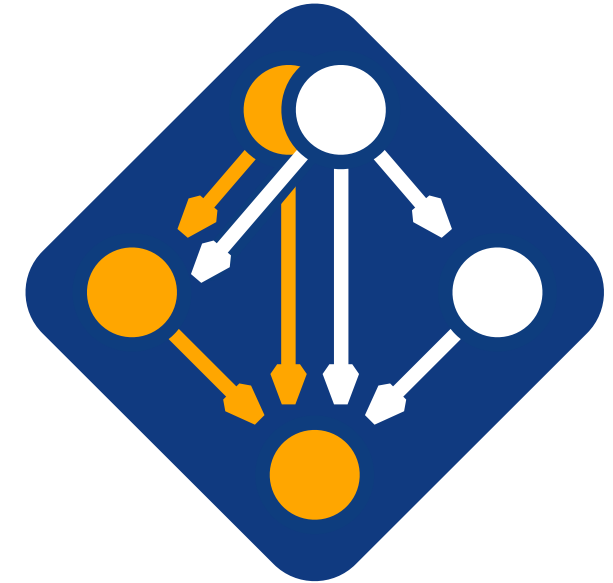
- How to install Spack (works out of the box):

```
$ git clone https://github.com/spack/spack  
$ . spack/share/spack/setup-env.sh
```

- How to install a package:

```
$ spack install tau
```

- TAU and its dependencies are installed within the Spack directory.
- Unlike typical package managers, Spack can also install many variants of the same build.
 - Different compilers
 - Different MPI implementations
 - Different build options



Visit spack.io

 github.com/spack/spack



HPSF
HIGH PERFORMANCE
SOFTWARE FOUNDATION

Spack provides the *spec* syntax to describe custom configurations

```
$ git clone https://github.com/spack/spack  
$ . spack/share/spack/setup-env.sh  
$ spack compiler find  
$ spack external find
```

set up compilers

set up external packages

```
$ spack install tau  
$ spack install tau@2.34.1  
$ spack install tau@2.34.1 %gcc@12.4.0  
$ spack install tau@2.34.1 %gcc@12.4.0 +rocm  
$ spack install tau@2.34.1 %gcc@12.4.0 +mpi ^mvapich2@4.0
```

unconstrained

@ custom version

% custom compiler

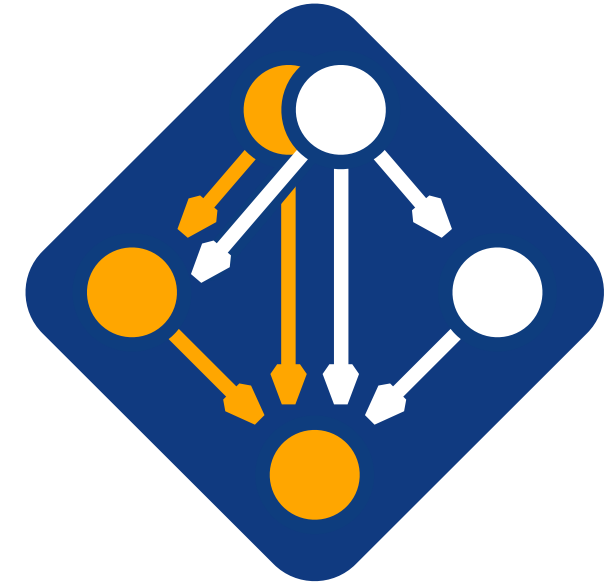
+/- build option

^ dependency information

- Each expression is a ***spec*** for a particular configuration
 - Each clause adds a constraint to the spec
 - Constraints are optional – specify only what you need.
 - Customize install on the command line!
- Spec syntax is recursive
 - Full control over the combinatorial build space

The Spack community is growing rapidly

- **Spack simplifies HPC software for:**
 - Users
 - Developers
 - Cluster installations
 - The largest HPC facilities
- **Spack is central to HPSF's software strategy**
 - Enable software reuse for developers and users
 - Allow the facilities to consume the entire E4S
- **The roadmap is packed with new features:**
 - Building the software distribution
 - Better workflows for building containers
 - Stacks for facilities
 - Chains for rapid dev workflow
 - Optimized binaries
 - Better dependency resolution



Visit spack.io
hpsf.io



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-4.2.3-47excoypwhfmx57rfs6reouvnnugcf)

[+] /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-feuxx36lsqp7quzmhmo4opbiadwpsars [6/6]

==> No binary for valgrind-3.23.0-feuxx36lsqp7quzmhmo4opbiadwpsars 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-feuxx36lsqp7quzmhmo4opbiadwpsars

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-feuxx36lsqp7quzmhmo4opbiadwpsars

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-feuxx36lsqp7quzmhmo4opbiadwpsars/bin/valgrind

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-feuux36lsqp7quzmhmo4opbiadwpsars/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.
-s                      same as --show-error-list=yes
```

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 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

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       petsc@3.22.4       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          raja@2024.07.0     umpire@2024.07.0
caliper@2.12.1     flux-core@0.67.0  kokkos@4.6.01      mfem@4.7.0           slate@2024.10.29    umpire@2024.07.0
camp@2024.07.0     ginkgo@1.9.0      kokkos@4.6.01      mgard@2023-12-09     slepc@3.22.2        umpire@2024.07.0

==> 57 installed packages
Singularity> nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2025 NVIDIA Corporation
Built on Fri_Feb_21_20:23:50_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   Perf          Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
|====+=====+=====+=====+=====+=====+=====+
| 0   NVIDIA H100 PCIe      On          | 00000000:E1:00:00 Off |                    0 |
```



NVIDIA® BioNeMo™ Framework on E4S 25.06 CUDA x86_64

The screenshot displays a JupyterLab interface within a Firefox browser window. The terminal at the top shows the following commands and output:

```
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.
```

The main notebook window, titled 'zeroshot_brca1 — Mozilla Firefox (on illyad)', shows the 'Zero-shot prediction of BRCA1 variant effects with Evo 2' tutorial. The tutorial text includes:

- 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.

The notebook contains two code cells:

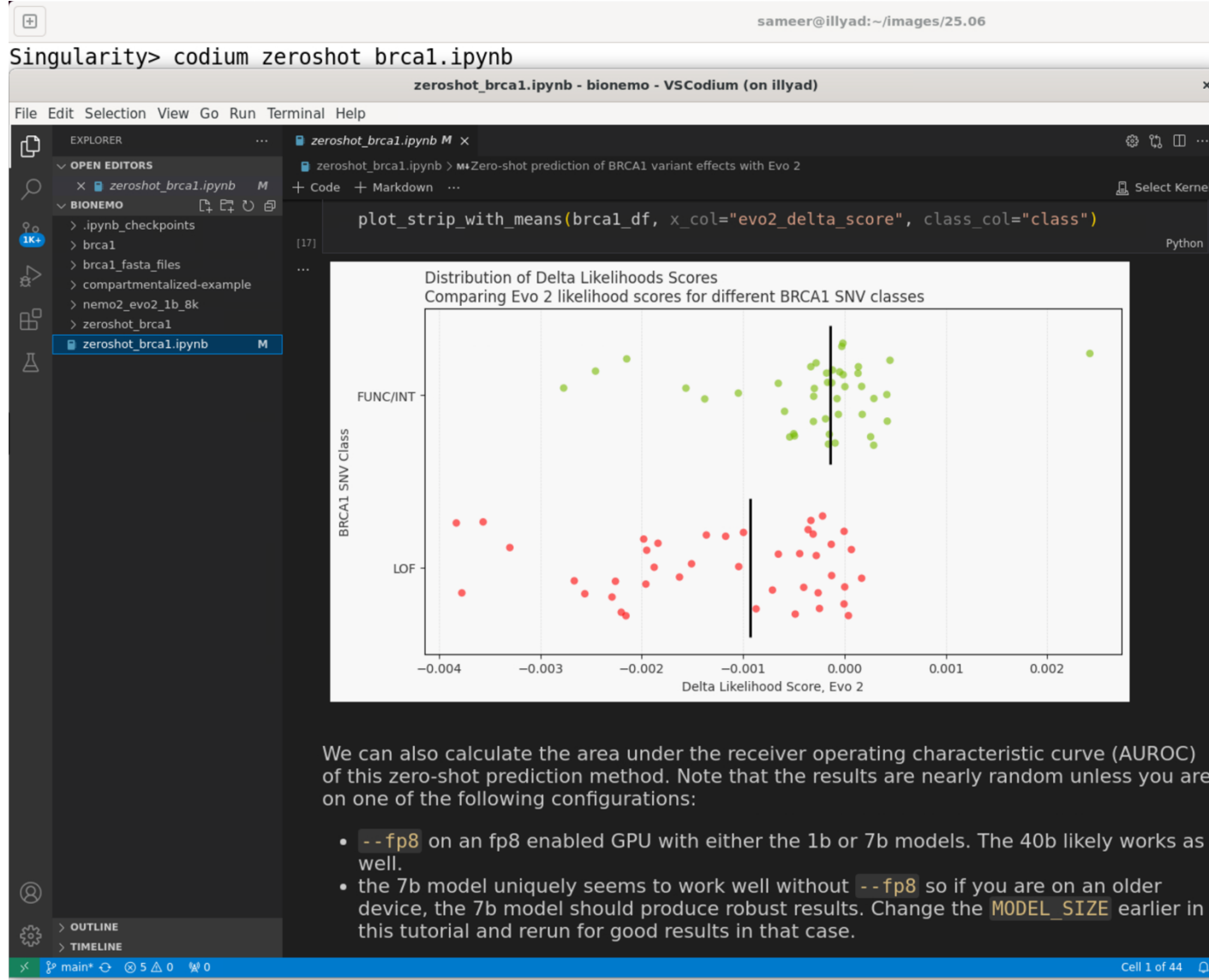
```
[1]: %%capture
import os

# 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
```

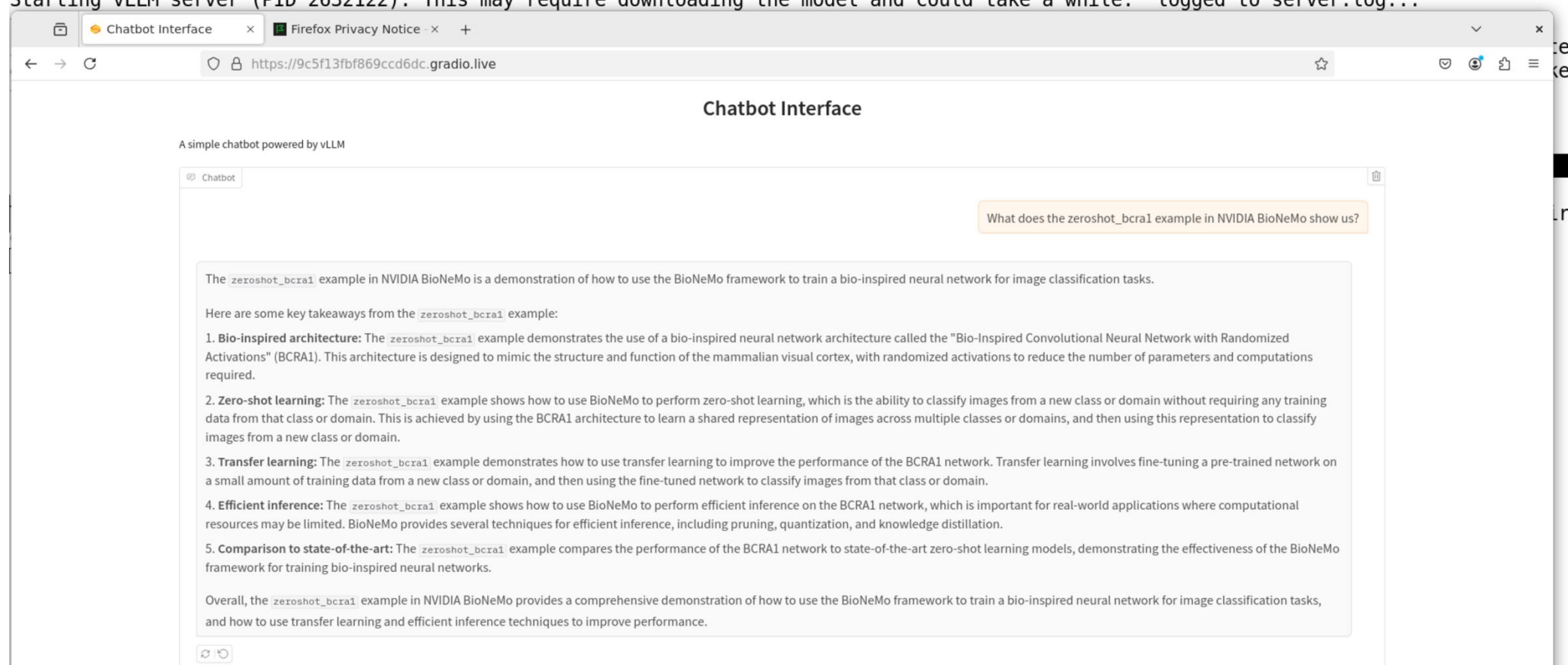
E4S 25.06 image for CUDA and x86_64 with VSCodium IDE



- 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

```
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...
```



Chatbot Interface

A simple chatbot powered by vLLM

Chatbot

What does the zeroshot_bcra1 example in NVIDIA BioNeMo show us?

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 terminal window and the ParaView 5.13.3 application. The terminal shows the execution of a Singularity command to run a GPU-accelerated application. The output indicates that the application is running synchronized to the vertical refresh rate and provides 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 ParaView interface shows a 3D visualization of a fluid flow simulation. A blue cylindrical object is shown with a red and white internal structure. The 'RenderView1' window displays a color scale for 'Pres' (Pressure) ranging from 2.029e-02 to 2.828e-02 and 'Temp' (Temperature) ranging from 400 to 9.132e+02. The 'Pipeline Browser' on the left shows the loaded data sources and filters, including 'disk_out_ref.ex2', 'Clip1', 'StreamTracer1', 'Tube1', and 'Glyph1'. The 'Properties' panel for 'Tube1' shows settings for 'AngularVelocity', 'Normals', 'Number of Sides', 'Capping', 'Radius', 'Vary Radius', and 'Radius Factor'.

```
sameer@mothra: ~/images
sameer@mothra:~/images$ singularity run --nv e4s-cuda80-x86_64-25.06.sif
Singularity> glxgears
Running synchronized to the vertical refresh. The framerate should be
approximately the same as the monitor refresh rate.
22335 frames in 5.0 seconds = 4466.834 FPS
24347 frames in 5.0 seconds = 4869.266 FPS
22561 frames in 5.0 seconds = 4512.186 FPS
^C
Singularity> paraview &
```

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

ParaTools Pro for E4S™: A cloud image for tool interoperability

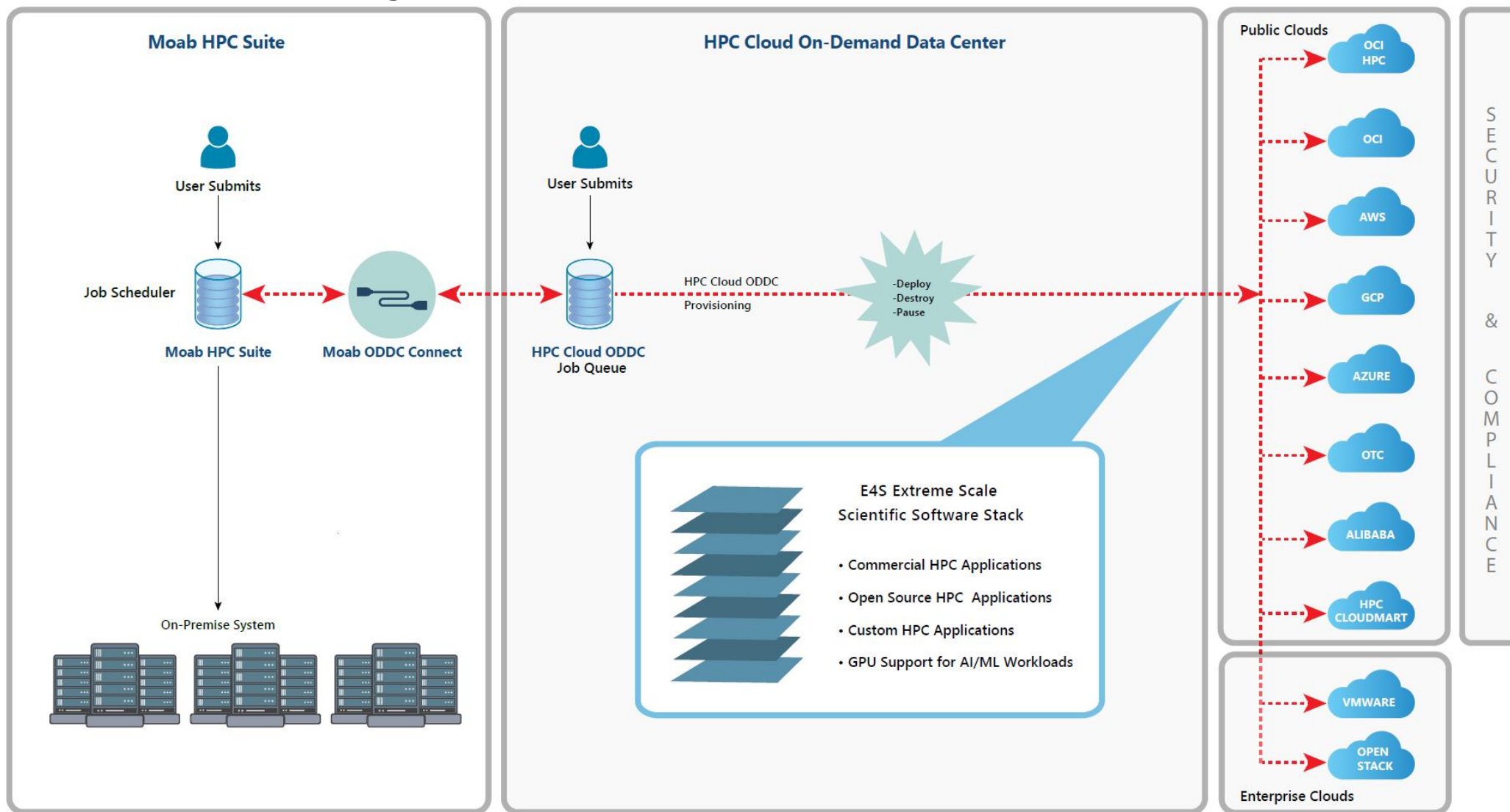
<https://paratoolspro.com>



Key considerations for cloud-based deployment for E4S

- MPI - the core inter-node communication library has several implementations
 - Intel MPI, MVAPICH2-X, OpenMPI
 - Interfacing MPI with the job scheduling package (MOAB, Torque, SLURM)
- Cloud providers have different inter-node network adapters:
 - Elastic Fabric Adapter (EFA) on AWS
 - Infiniband on Azure
 - Mellanox Connect-X 5 Ethernet (ROCE) on Oracle Cloud Infrastructure (OCI)
 - IPU on Google Cloud (GCP)
- Intra-node communication with XPMEM (driver and kernel module support is critical)
- GPU Direct Async (GDR) support for communication between GPUs in MVPICH-Plus release
- ParaTools, Inc. building E4S optimized with MVAPICH-Plus for AWS, OCI, GCP, and Azure
- Using Adaptive Computing, Inc.'s Heidi/ODDC to launch E4S jobs on multiple cloud providers!

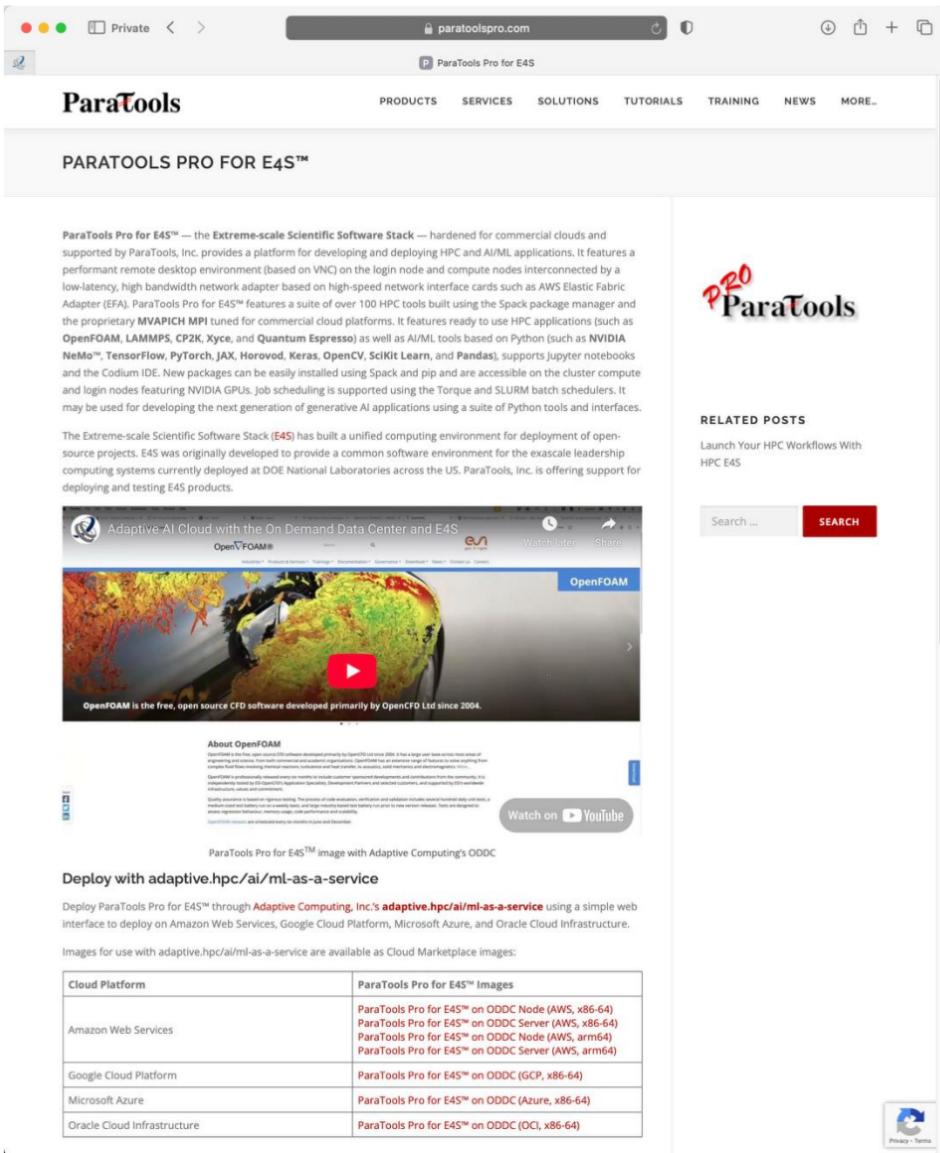
Adaptive Computing's Heidi/ODDC with ParaTools Pro for E4S™



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

U.S. DEPARTMENT OF
ENERGY

Office of
Science



The screenshot shows the ParaTools Pro for E4S website. The header includes the ParaTools logo and navigation links: PRODUCTS, SERVICES, SOLUTIONS, TUTORIALS, TRAINING, NEWS, MORE... The main heading is "PARATOOLS PRO FOR E4S™". Below it, a paragraph describes ParaTools Pro for E4S as an Extreme-scale Scientific Software Stack hardened for commercial clouds, supporting various HPC and AI/ML applications. It lists supported tools like OpenFOAM, LAMMPS, CP2K, Xyce, Quantum Espresso, NVIDIA NeMo, TensorFlow, PyTorch, JAX, Horovod, Keras, OpenCV, SciKit Learn, and Pandas. A video player titled "Adaptive AI Cloud with the On Demand Data Center and E4S" is shown, featuring a visualization of a molecular structure. To the right, there's a section for "RELATED POSTS" with a link to "Launch Your HPC Workflows With HPC E4S". At the bottom, a table lists cloud platforms and their corresponding ParaTools Pro for E4S 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)



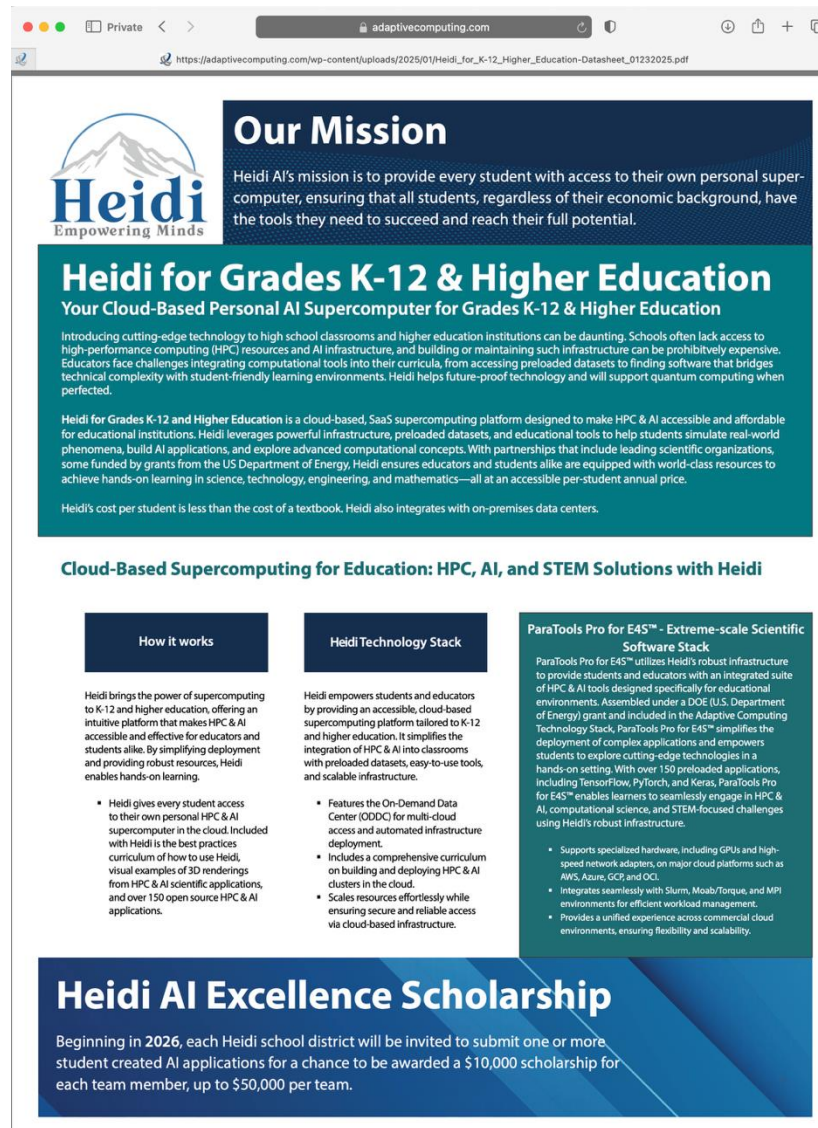
U.S. DEPARTMENT OF
ENERGY

Office of
Science

Office of
Science

- 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', describing it as a cloud-based personal AI supercomputer. It mentions that Heidi helps future-proof technology and will support quantum computing. Below this, it states that 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. It 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.

Below the main text is a section titled 'Cloud-Based Supercomputing for Education: HPC, AI, and STEM Solutions with Heidi'. It contains three columns of information:

- How it works**: 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 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.
- Heidi Technology Stack**: 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.
 - 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.
- ParaTools Pro for E4S™ - Extreme-scale Scientific Software Stack**: 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.
 - 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.

At the bottom is the 'Heidi AI Excellence Scholarship' section, stating 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.

- 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

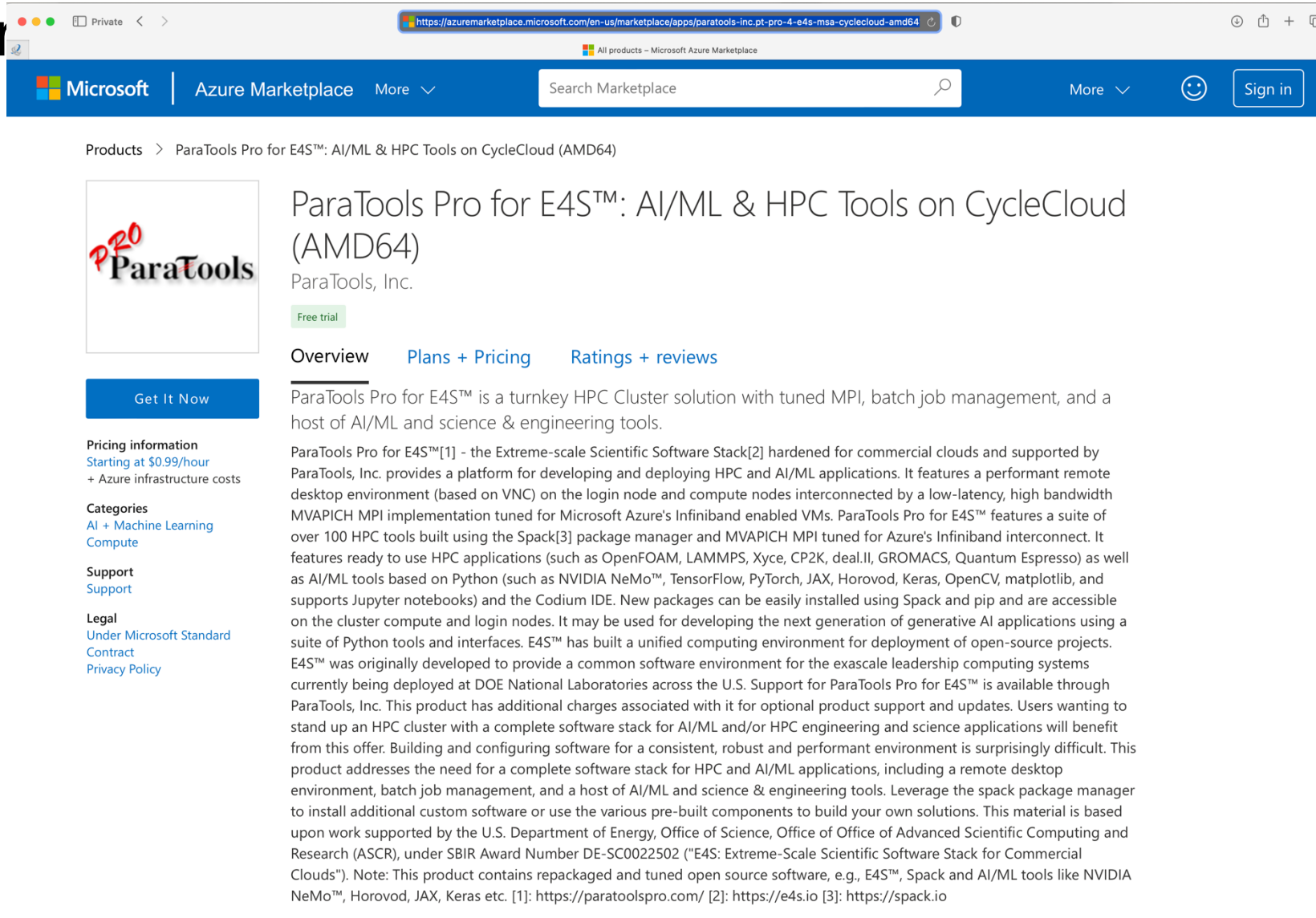
The screenshot shows the AWS Marketplace search results for "ParaTools Pro for E4S". The search bar at the top contains "ParaTools Pro for E4S". The results are filtered to show 8 results. The left sidebar contains filters for Categories (Infrastructure Software (8), Machine Learning (8), Industries (8)), Delivery methods (Amazon Machine Image (8)), Publisher (ParaTools Inc. (8)), Pricing model (Usage Based (8)), Operating system (All Linux/Unix), Free trial (Free Trial (8)), Contract type (Standard Contract (8)), Architecture (64-bit (x86) (4), 64-bit (Arm) (4)), and Region (Africa (Cape Town) (8), Asia Pacific (Hong Kong) (8), Asia Pacific (Tokyo) (8), Asia Pacific (Seoul) (8), Asia Pacific (Osaka) (8), Asia Pacific (Mumbai) (8), Asia Pacific (Hyderabad) (8), Asia Pacific (Singapore) (8), Asia Pacific (Sydney) (8), Asia Pacific (Jakarta) (8)). The main content area displays four results, all by ParaTools Inc. Each result includes a title, version, and a description. The results are: 1. ParaTools Pro for E4S™: AI/ML & HPC Tools on ParallelCluster (arm64), 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. 2. ParaTools Pro for E4S™: AI/ML & HPC Tools on ParallelCluster (x86), 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. 3. ParaTools Pro for E4S™: AI/ML & HPC Tools on AWS PCS (x86), 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. 4. ParaTools Pro for E4S™: AI/ML & HPC Tools on ODDC Node (x86), Ver v17423266-odd-cv1.0-e4s-24.11-node-amd64, Free Trial, Starting from \$0.99 to \$0.99/hr for software + AWS usage fees.

ParaTools Pro for E4S™
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

Mar



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.

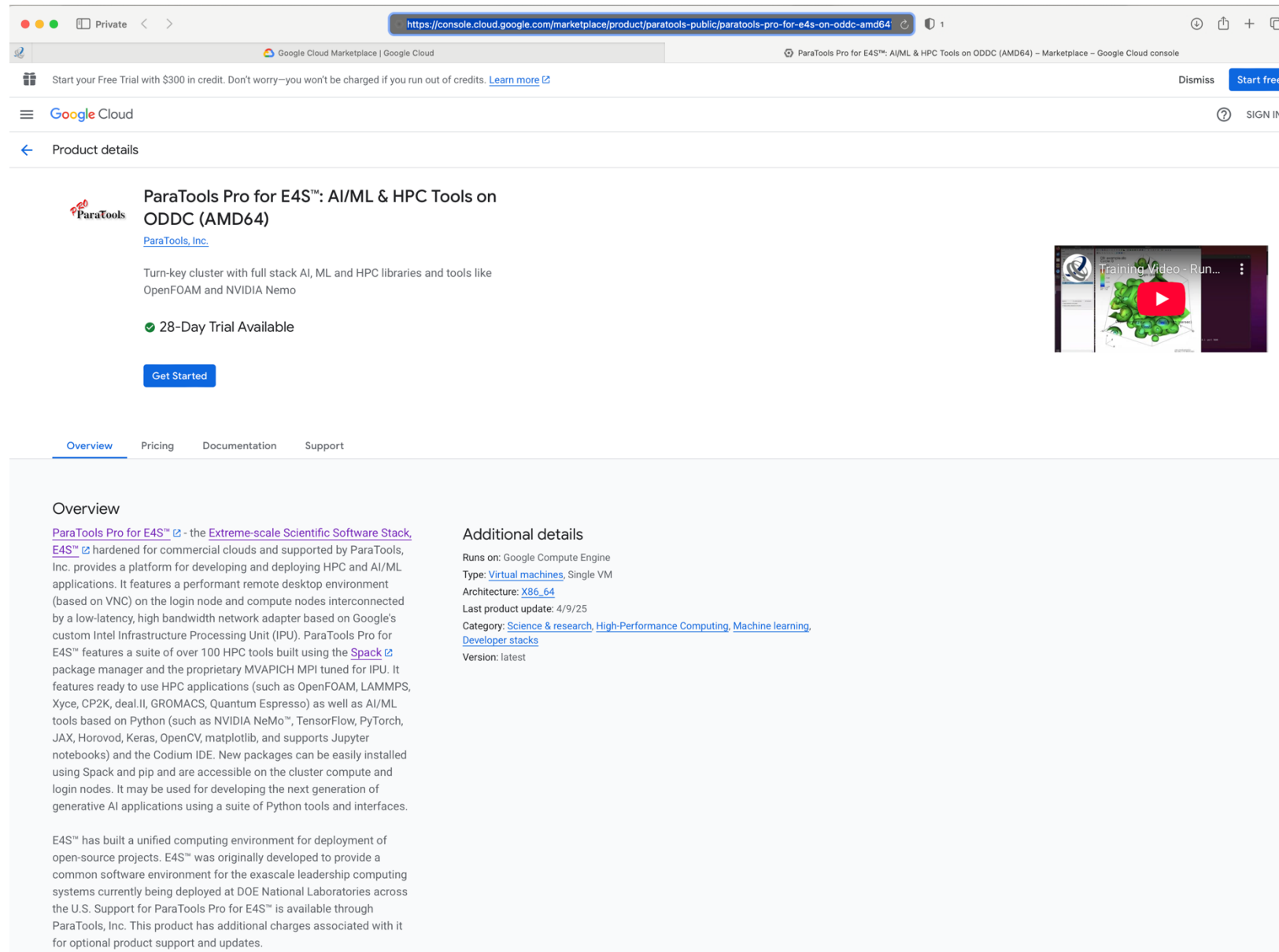
ParaTools Pro for E4S™[1] - the Extreme-scale Scientific Software Stack[2] hardened for commercial clouds and supported by ParaTools, Inc. provides a platform for developing and deploying HPC and AI/ML applications. It features a performant remote desktop environment (based on VNC) on the login node 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 MVAPICH 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

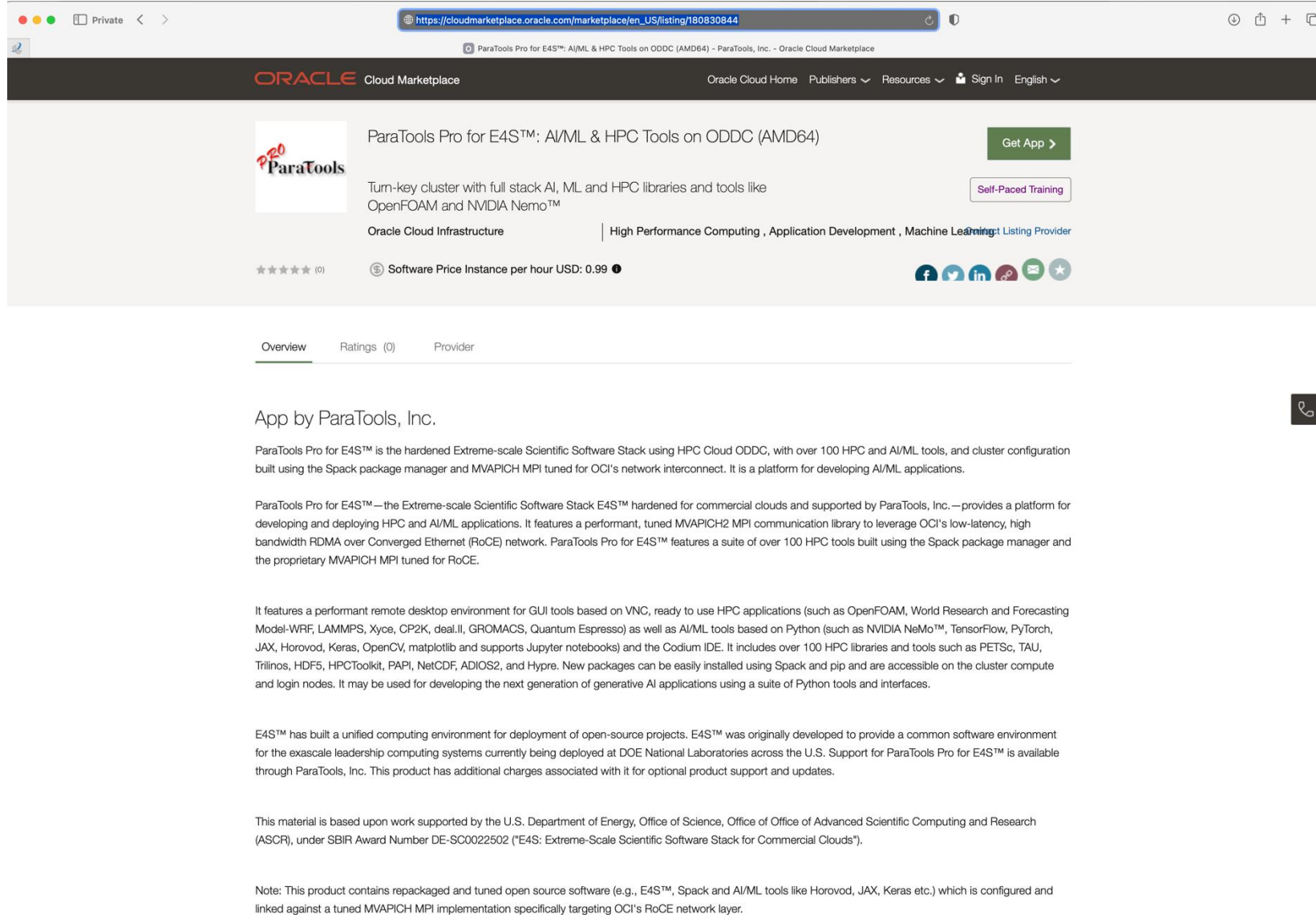
Runs on: Google Compute Engine
Type: [Virtual machines](#), Single VM
Architecture: [X86_64](#)
Last product update: 4/9/25
Category: [Science & research](#), [High-Performance Computing](#), [Machine learning](#), [Developer stacks](#)
Version: latest

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



Oracle Cloud Marketplace

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

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

Oracle Cloud Infrastructure | High Performance Computing , Application Development , Machine Learning | Listing Provider

★★★★★ (0) Software Price Instance per hour USD: 0.99

Overview Ratings (0) Provider

App by ParaTools, Inc.

ParaTools Pro for E4S™ is the hardened Extreme-scale Scientific Software Stack using HPC Cloud ODDC, with over 100 HPC and AI/ML tools, and cluster configuration built using the Spack package manager and MVAPICH MPI tuned for OCI's network interconnect. It is a platform for developing AI/ML applications.

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, tuned MVAPICH2 MPI communication library to leverage OCI's low-latency, high bandwidth RDMA over Converged Ethernet (RoCE) network. 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 RoCE.

It features a performant remote desktop environment for GUI tools based on VNC, ready to use HPC applications (such as OpenFOAM, World Research and Forecasting Model-WRF, 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. It includes over 100 HPC libraries and tools such as PETSc, TAU, Trilinos, HDF5, HPCToolkit, PAPI, NetCDF, ADIOS2, and Hypr. 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.

This material is based upon work supported by the U.S. Department of Energy, Office of Science, 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 Horovod, JAX, Keras etc.) which is configured and linked against a tuned MVAPICH MPI implementation specifically targeting OCI's RoCE network layer.

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

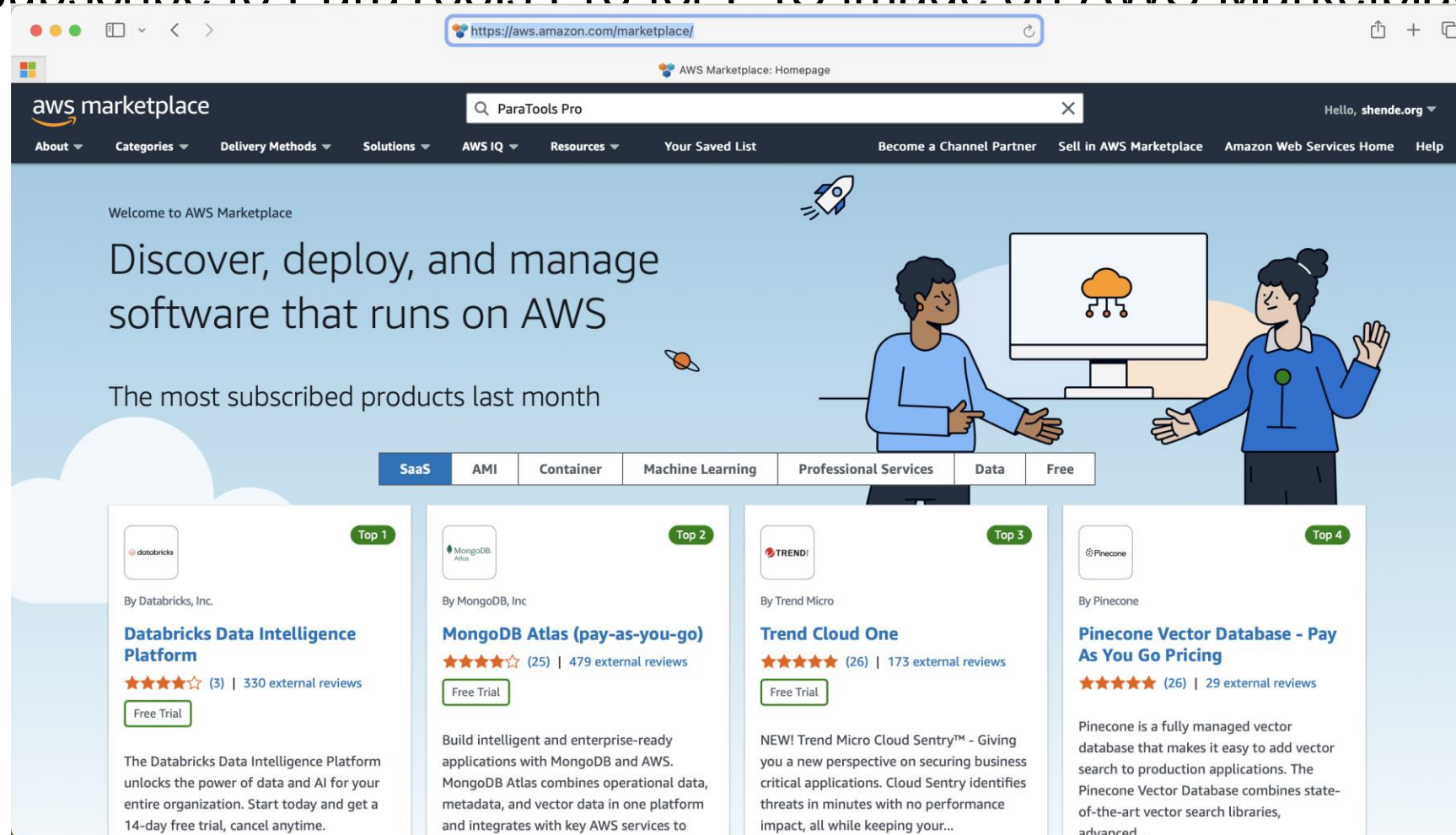
E4S: An HPC-AI Software Ecosystem for Science!

The screenshot displays a remote desktop environment with the following components:

- Terminal Window:** Shows the Singularity prompt and a list of installed packages. The command `spack find +cuda` has been executed, resulting in 57 installed packages. The packages are listed in a grid format, including `adios2@2.10.2`, `amrex@25.03`, `arborx@1.5`, `axom@0.10.1`, `blaspp@2024.10.26`, `bricks@2023.08.25`, `cabana@0.7.0`, `caliper@2.12.1`, `camp@2024.07.0`, `chai@2024.07.0`, `chapel@2.4.0`, `cp2k@2025.1`, `cusz@0.14.0`, `fftx@1.2.0`, `flecsi@2.3.2`, `flux-core@0.67.0`, `ginkgo@1.9.0`, `gromacs@2024.4`, `heffte@2.4.1`, `hpctoolkit@2024.01.1`, `hpx@1.10.0`, `hwloc@2.11.1`, `hypre@2.32.0`, `kokkos@4.5.01`, `kokkos@4.6.01`, `kokkos@4.6.01`, `kokkos@4.6.01`, `kokkos-kernels@4.6.01`, `lammps@20240829.1`, `lapackpp@2024.10.26`, `legion@24.12.0`, `libceed@0.12.0`, `magma@2.9.0`, `mfem@4.7.0`, `mgard@2023-12-09`, `nvcomp@2.2.0`, `papi@7.1.0`, `parsec@3.0.2209`, `petsc@3.22.4`, `petsc@3.22.4`, `raja@2024.07.0`, `raja@2024.07.0`, `slate@2024.10.29`, `slepc@3.22.2`, `strumpack@8.0.0`, `sundials@7.2.1`, `superlu-dist@9.1.0`, `tasmanian@8.1`, `tau@2.34.1`, `trilinos@16.1.0`, `umpire@2024.07.0`, `umpire@2024.07.0`, `umpire@2024.07.0`, `upcxx@2023.9.0`, `vtm@2.2.0`, and `zfp@1.0.0`.
- VSCode Editor:** The file `foo.py` is open, showing a Python script that converts .sph files to .wav files using `sox` and processes them with the `nemo` package for ASR. The script includes comments about the NeMo's "core" package and the QuartzNet model.
- 3D Visualizer:** The TAU: ParaProf: 3D Visualizer window displays a 3D surface plot of a dataset. The plot shows a complex, multi-peaked surface with a color scale ranging from 0 to 365.836 seconds. The axes are labeled "Time" and "Space". Below the plot, there is a table showing the minimum values for four nodes (node 0, node 1, node 2, node 3) across different categories (Min, node 0, node 1, node 2, node 3).

Using your own AWS credentials with Adaptive Computing's Heidi

STEP 1: Subscribe to ParaTools Pro for F4S image on AWS Marketplace



Go to <https://aws.amazon.com/marketplace>
Login, then search for ParaTools Pro

STEP 1: Subscribe to ParaTools Pro for E4S image on AWS Marketplace

aws marketplace

ParaTools Pro

Hello, shende.org

About Categories Delivery Methods Solutions AWS IQ Resources Your Saved List Become a Channel Partner Sell in AWS Marketplace Amazon Web Services Home Help

Amazon Machine Image (6)

▼ Publisher

- ☐ ParaTools Inc. (6)

▼ Pricing model

- ☐ Usage Based (6)

Operating system

- ☐ All Linux/Unix

▼ Free trial

- ☐ Free Trial (6)

▼ Contract type

- ☐ Standard Contract (6)

▼ Architecture

- ☐ 64-bit (x86) (3)
- ☐ 64-bit (Arm) (3)

▼ Region

- ☐ Africa (Cape Town) (6)
- ☐ Asia Pacific (Hong Kong) (6)
- ☐ Asia Pacific (Tokyo) (6)
- ☐ Asia Pacific (Seoul) (6)
- ☐ Asia Pacific (Osaka) (6)
- ☐ Asia Pacific (Mumbai) (6)
- ☐ Asia Pacific (Hyderabad) (6)
- ☐ Asia Pacific (Singapore) (6)
- ☐ Asia Pacific (Sydney) (6)
- ☐ Asia Pacific (Jakarta) (6)

Show 10 More

ParaTools Pro (6 results) showing 1 - 6

Did you mean [paratool pre?](#)

Sort By: Relevance

ParaTools

[ParaTools Pro for E4S™: AI/ML & HPC Tools on ODDC Node \(x86\)](#)

By [ParaTools Inc.](#) | Ver v17208158-oddc-v1.0-e4s-24.05-node-amd64

Free Trial

Starting from \$0.99 to \$0.99/hr for software + AWS usage fees

ParaTools Pro for E4S™[1] - the Extreme-scale Scientific Software Stack E4S[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...

ParaTools

[ParaTools Pro for E4S™: AI/ML & HPC Tools on ODDC Server \(x86\)](#)

By [ParaTools Inc.](#) | Ver 17208158-oddc-v1.0-e4s-24.05-server-amd64

Free Trial

Starting from \$0.99 to \$0.99/hr for software + AWS usage fees

ParaTools Pro for E4S™[1] - the Extreme-scale Scientific Software Stack E4S[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...

ParaTools

[ParaTools Pro for E4S™: AI/ML & HPC Tools on ParallelCluster \(x86\)](#)

By [ParaTools Inc.](#) | Ver v2024.07.18.0740-pcluster-3.10.1-e4s-24.05-amd64

Free Trial

Starting from \$0.99 to \$0.99/hr for software + AWS usage fees

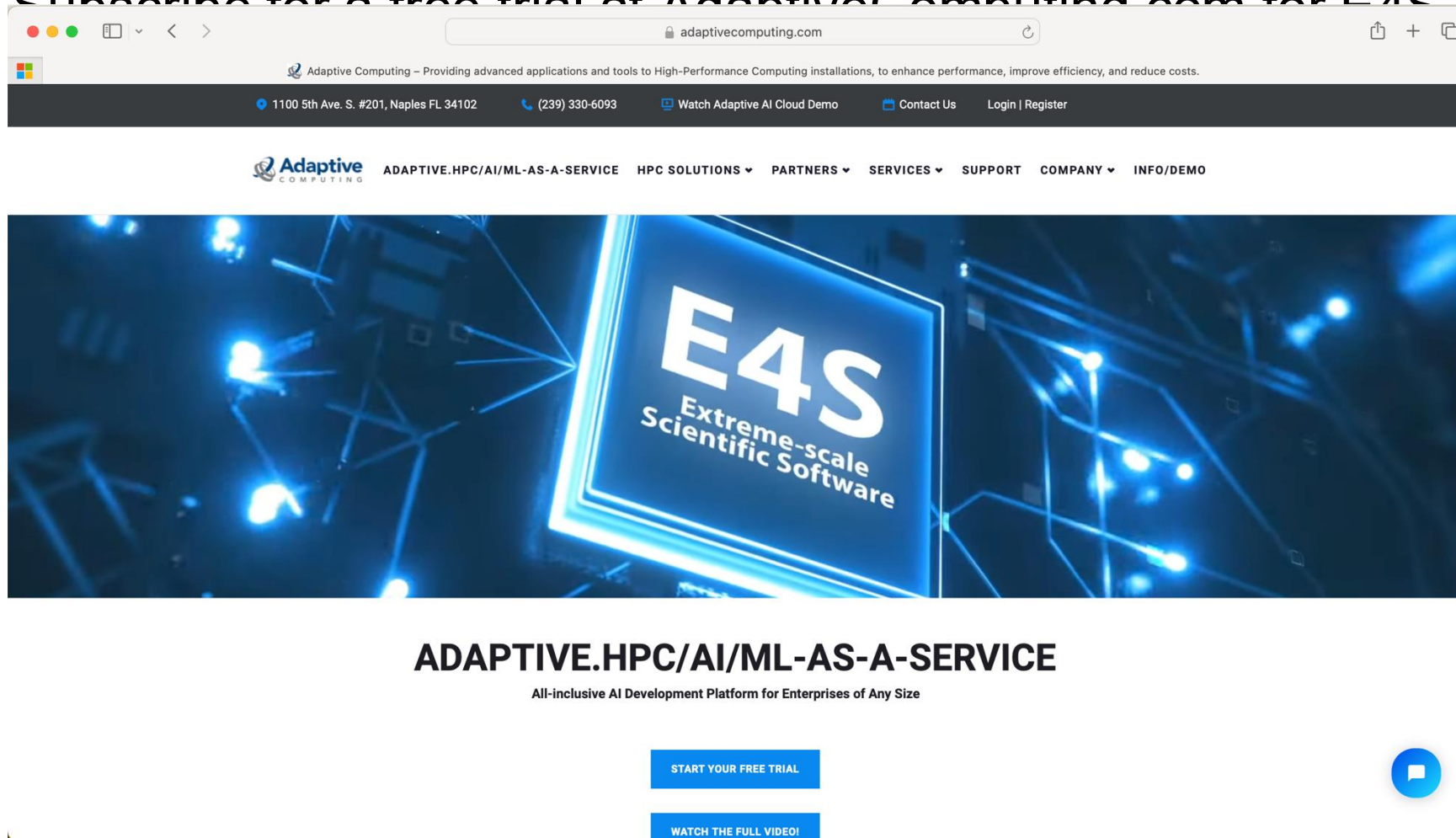
ParaTools Pro for E4S™[1] - the Extreme-scale Scientific Software Stack E4S[2] hardened for commercial clouds and supported by ParaTools, Inc. provides a platform for

Free trial for 31 days!

Click on ODDC Server (x86) click subscribe

Click on ODDC Node (x86) click subscribe

STEP 2: Subscribe for a free trial at AdaptiveComputing.com for E4S



Go to <https://adaptivecomputing.com>
Click on "Start your free trial"

STEP 2: Subscribe for a free trial on Adaptive Computing for E4S

adaptivecomputing.com

Request Trial – Adaptive Computing

1100 5th Ave. S. #201, Naples FL 34102 (239) 330-6093 Watch Adaptive AI Cloud Demo Contact Us Login | Register

Adaptive COMPUTING ADAPTIVE.HPC/AI/ML-AS-A-SERVICE HPC SOLUTIONS PARTNERS SERVICES SUPPORT COMPANY INFO/DEMO

REQUEST TRIAL

You Are Here: [HOME](#) / [REQUEST TRIAL](#)

Please fill out the form below to request a trial. A team member will contact you within 24 hours. Thank you for your interest in Adaptive Computing!

Full Name (required)

Email (required)

Phone Number (required)

Company Name (required)

Job Title (optional)

Please sign me up for the following trial(s)

☐ adaptive.ai-as-a-service ☐ ODDC (HPC On-Demand Data Center) ☒ E4S

I would like to:

☐ Request a sales call ☐ Get more information

Your Message (optional)

What else should we know?

Fill out the form and check the E4S box

Using ParaTools Pro for E4S™ image on AWS with Heidi AI/ODDC

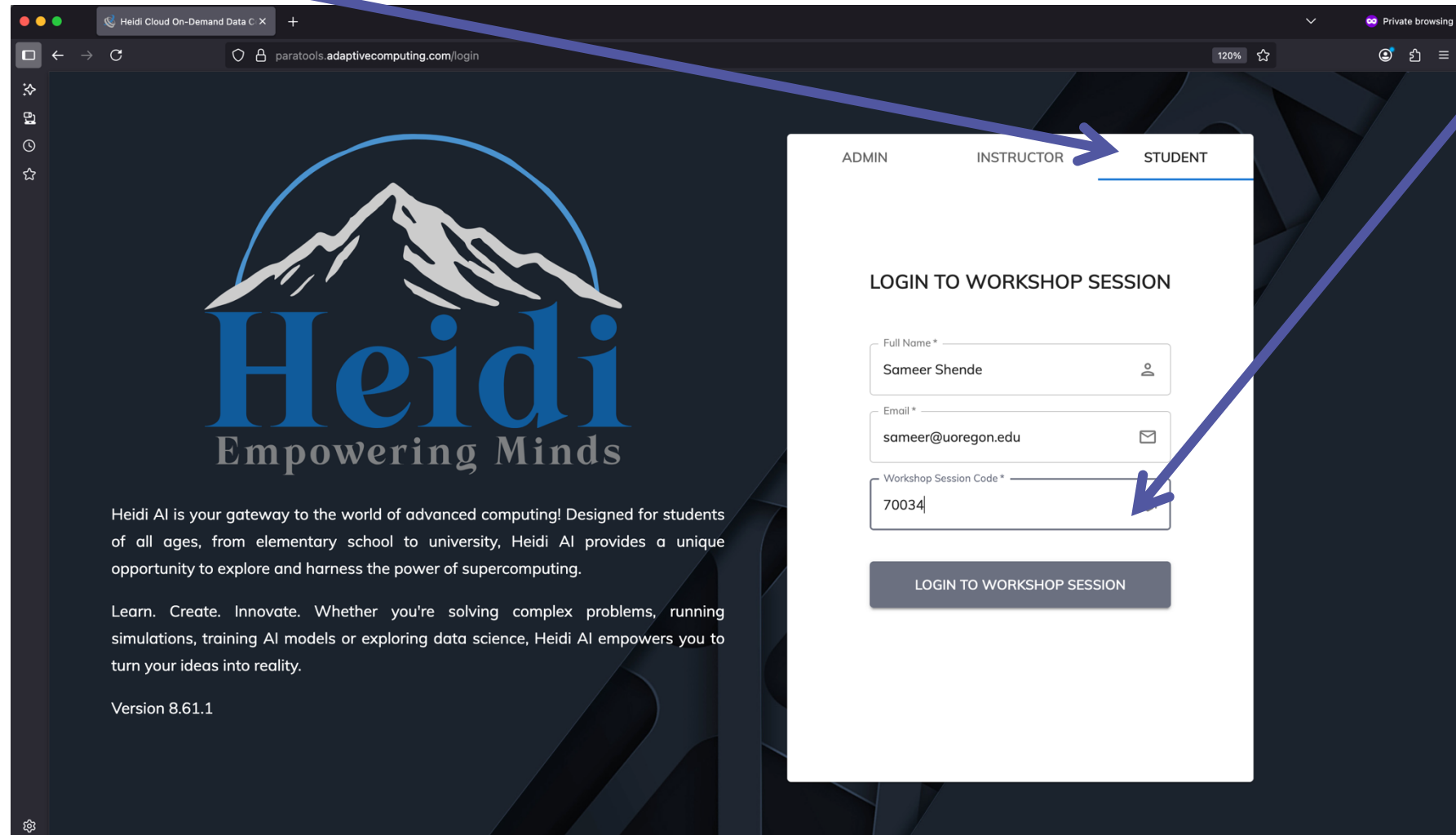
Login to:

<https://paratools.adaptivecomputing.com>

with the credentials. Firefox private window recommended.

Click on Student tab and use code:70034

- # Connect to <https://paratools.adaptivecomputing.com>
- Use Student tab and enter name, email, session code 70034



The screenshot shows a web browser window with the URL `paratools.adaptivecomputing.com/login`. The main page features the Heidi AI logo, which includes a mountain silhouette and the text "Heidi Empowering Minds". Below the logo, there is a paragraph about Heidi AI being a gateway to advanced computing, followed by a list of activities: "Learn. Create. Innovate. Whether you're solving complex problems, running simulations, training AI models or exploring data science, Heidi AI empowers you to turn your ideas into reality." and the version number "Version 8.61.1".

A modal form titled "LOGIN TO WORKSHOP SESSION" is overlaid on the right side of the page. It has three tabs: "ADMIN", "INSTRUCTOR", and "STUDENT". The "STUDENT" tab is selected. The form contains three input fields: "Full Name *" with the value "Sameer Shende", "Email *" with the value "sameer@uoregon.edu", and "Workshop Session Code *" with the value "70034". A "LOGIN TO WORKSHOP SESSION" button is at the bottom of the form. Two blue arrows point to the "STUDENT" tab and the "Workshop Session Code" input field.

ADMIN INSTRUCTOR **STUDENT**

LOGIN TO WORKSHOP SESSION

Full Name *
Sameer Shende

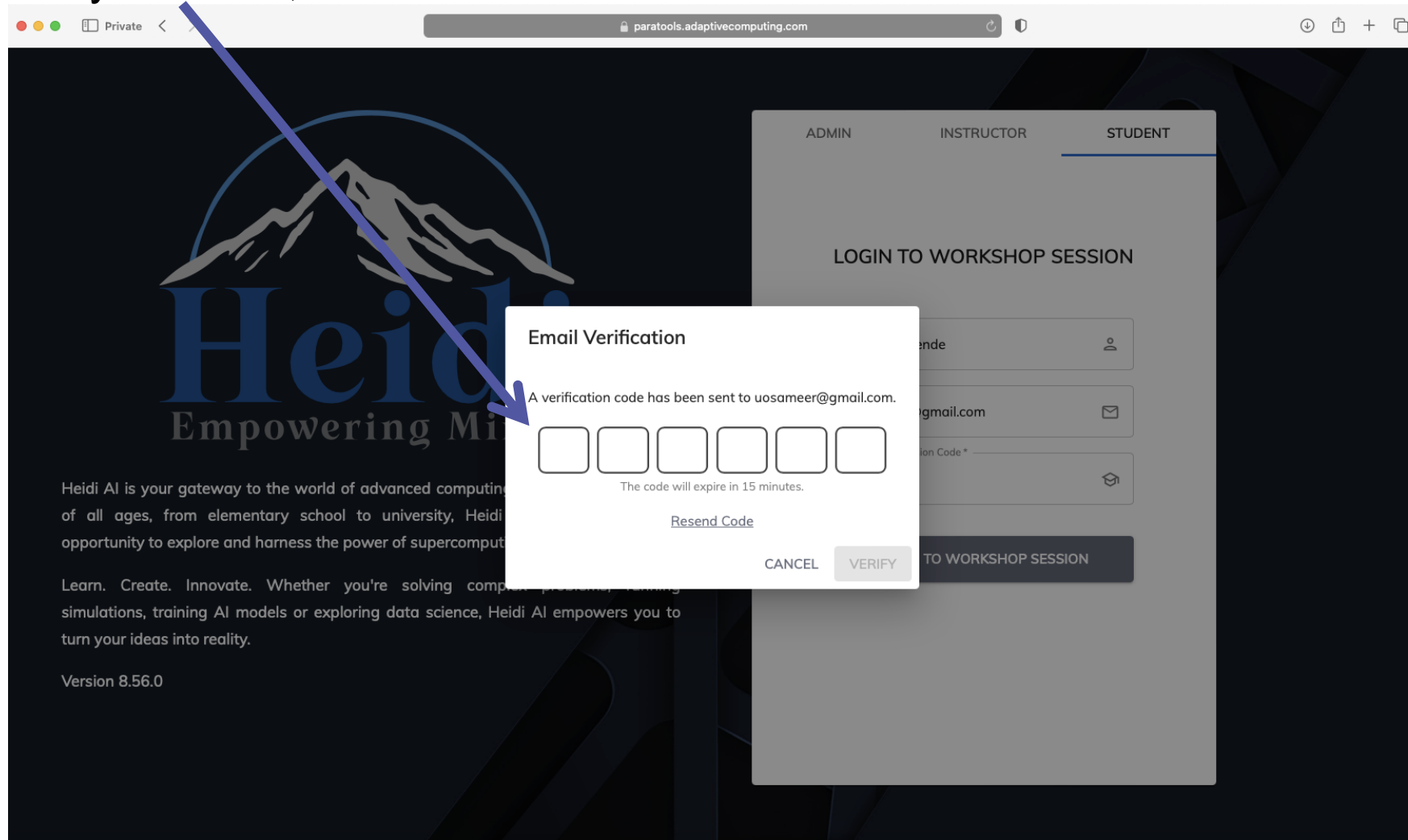
Email *
sameer@uoregon.edu

Workshop Session Code *
70034

LOGIN TO WORKSHOP SESSION

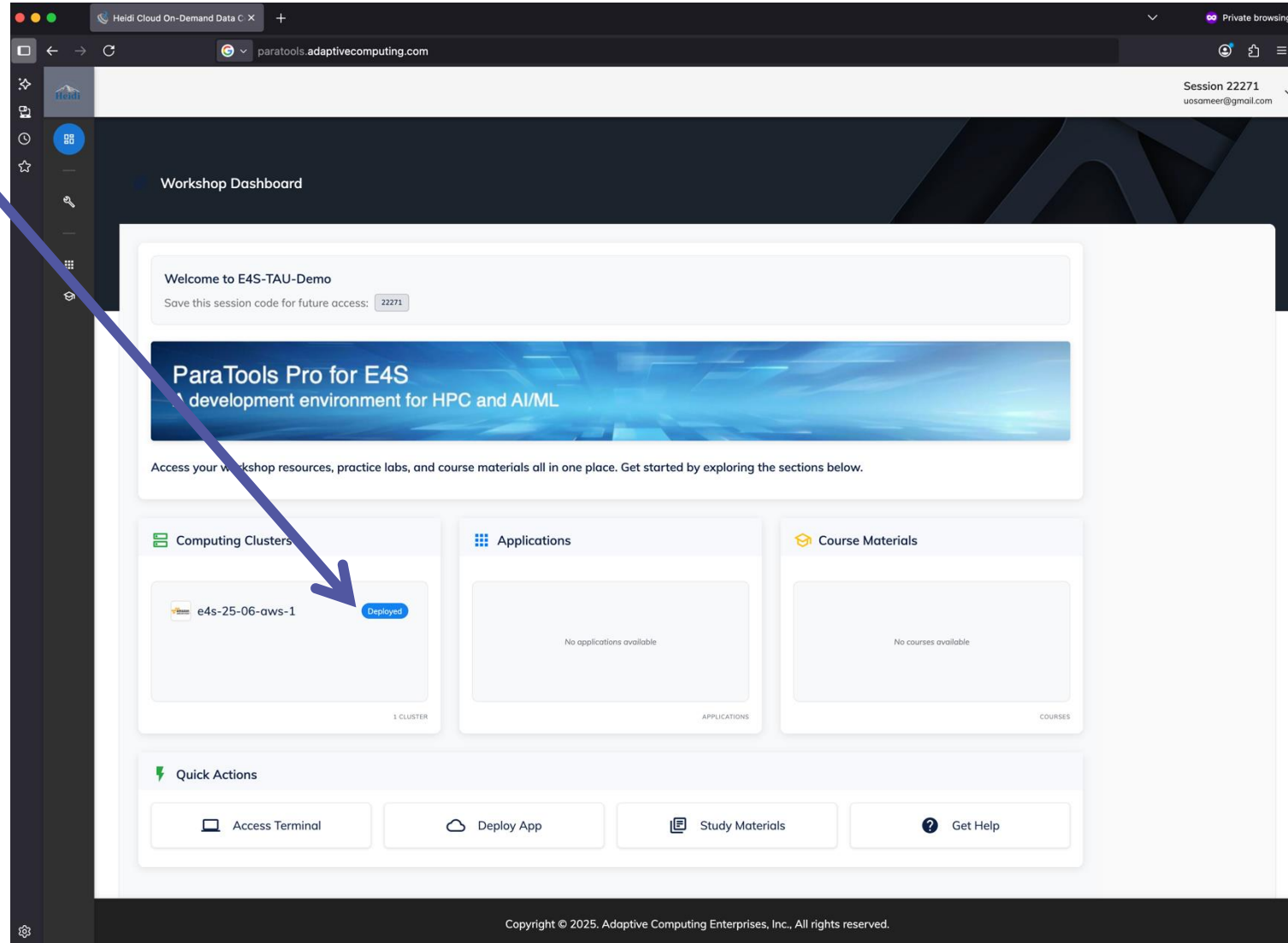
Connect to https://paratools.adaptivecomputing.com

- Check your email, enter verification code.



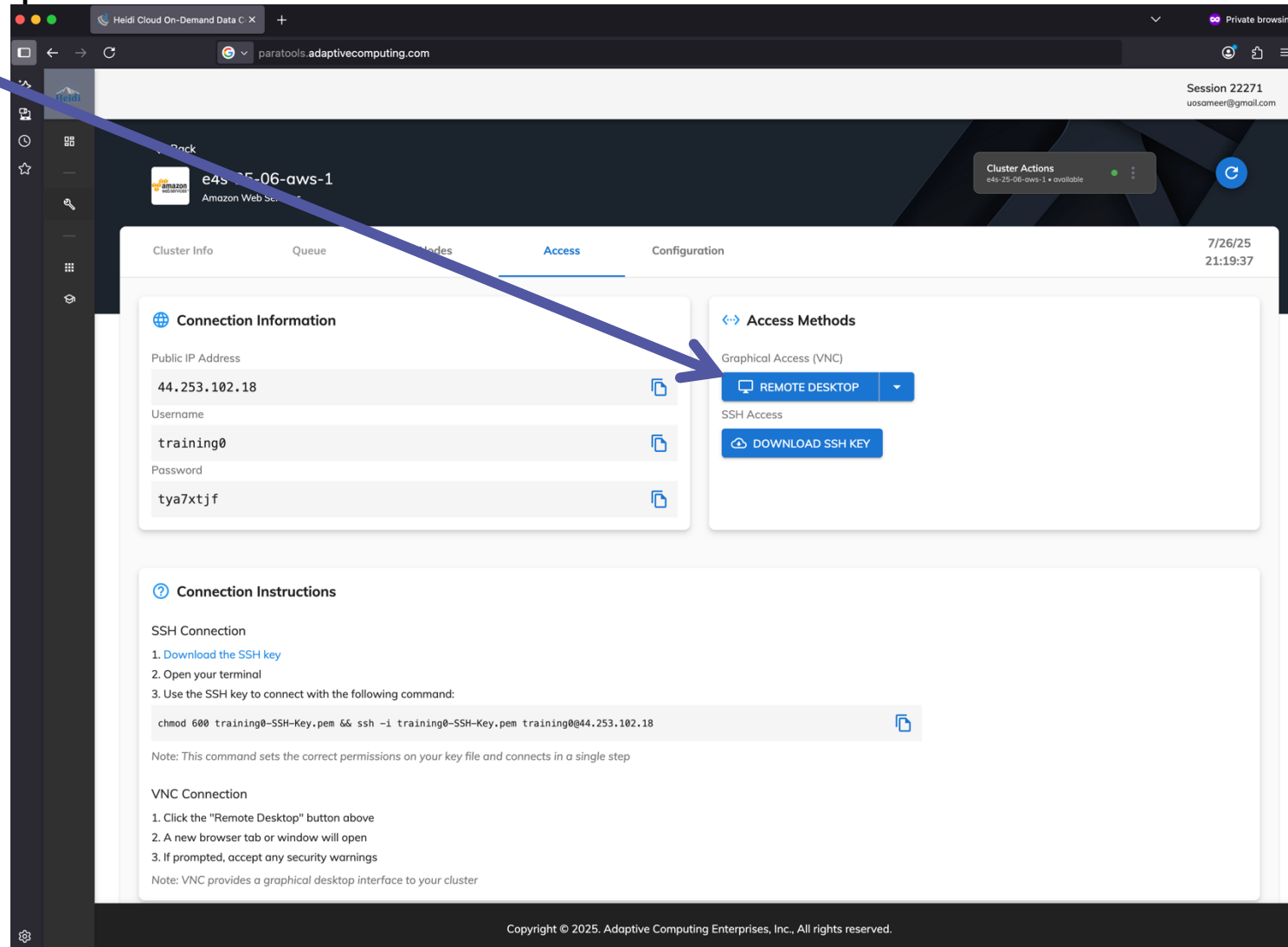
Connect to https://paratools.adaptivecomputing.com

- Click cluster



Connect to https://paratools.adaptivecomputing.com

- Click Remote Desktop



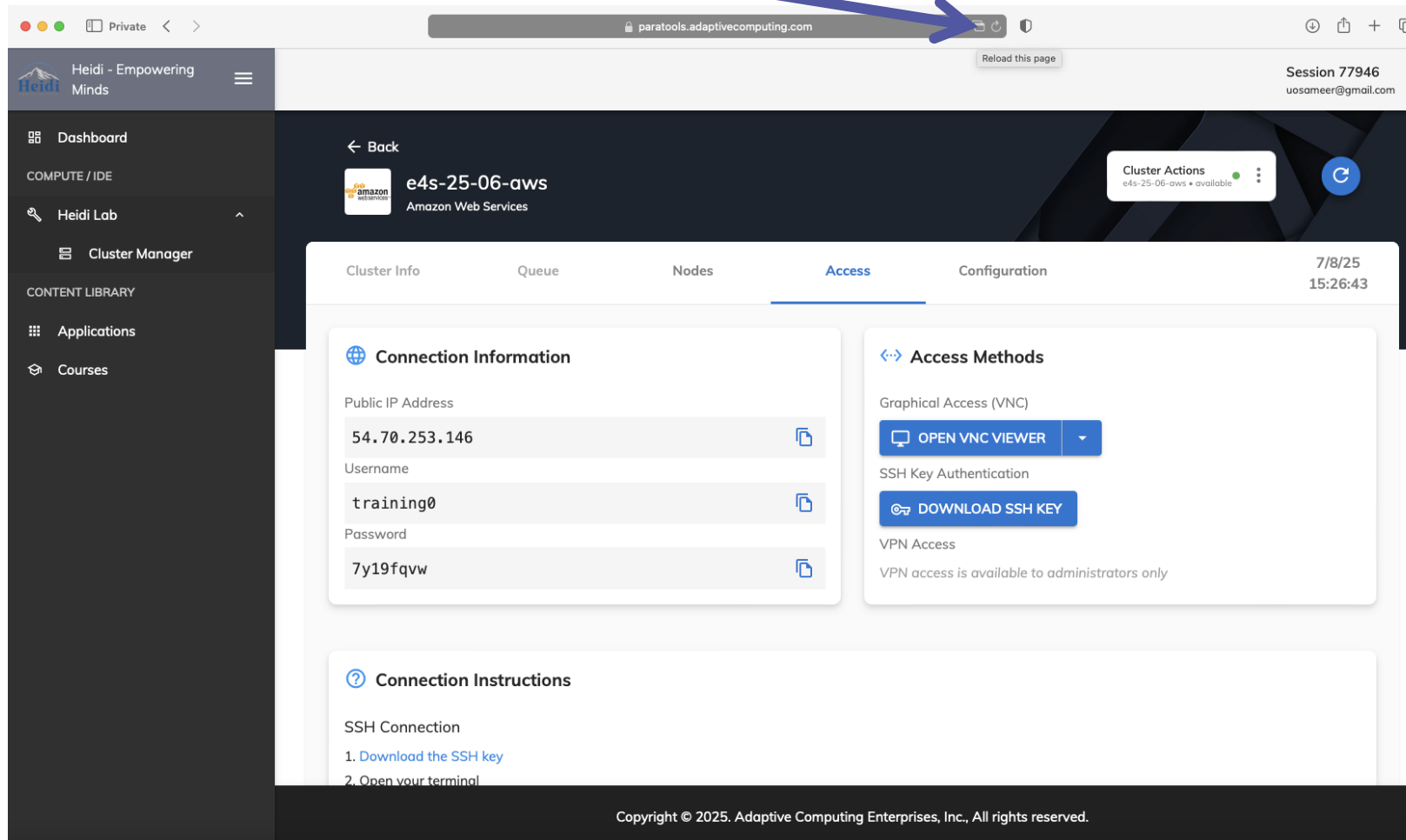
The screenshot shows the web interface of paratools.adaptivecomputing.com. The browser address bar displays the URL. The page header includes a session ID (22271) and a user email (uosameer@gmail.com). The main content area is divided into several sections:

- Cluster Info**: Shows the cluster name "e4s-25-06-aws-1" and the provider "Amazon Web Services".
- Access**: The active tab, containing:
 - Connection Information**: Fields for Public IP Address (44.253.102.18), Username (training0), and Password (tya7xtjf).
 - Access Methods**: Options for Graphical Access (VNC) and SSH Access. The "REMOTE DESKTOP" button is highlighted with a blue arrow.
 - Connection Instructions**: A section with steps for SSH and VNC connections, including a terminal command and a note about security warnings.

The footer contains the copyright notice: "Copyright © 2025, Adaptive Computing Enterprises, Inc., All rights reserved."

Connect to <https://paratools.adaptivecomputing.com>

- You may have to enable pop-up windows and accept

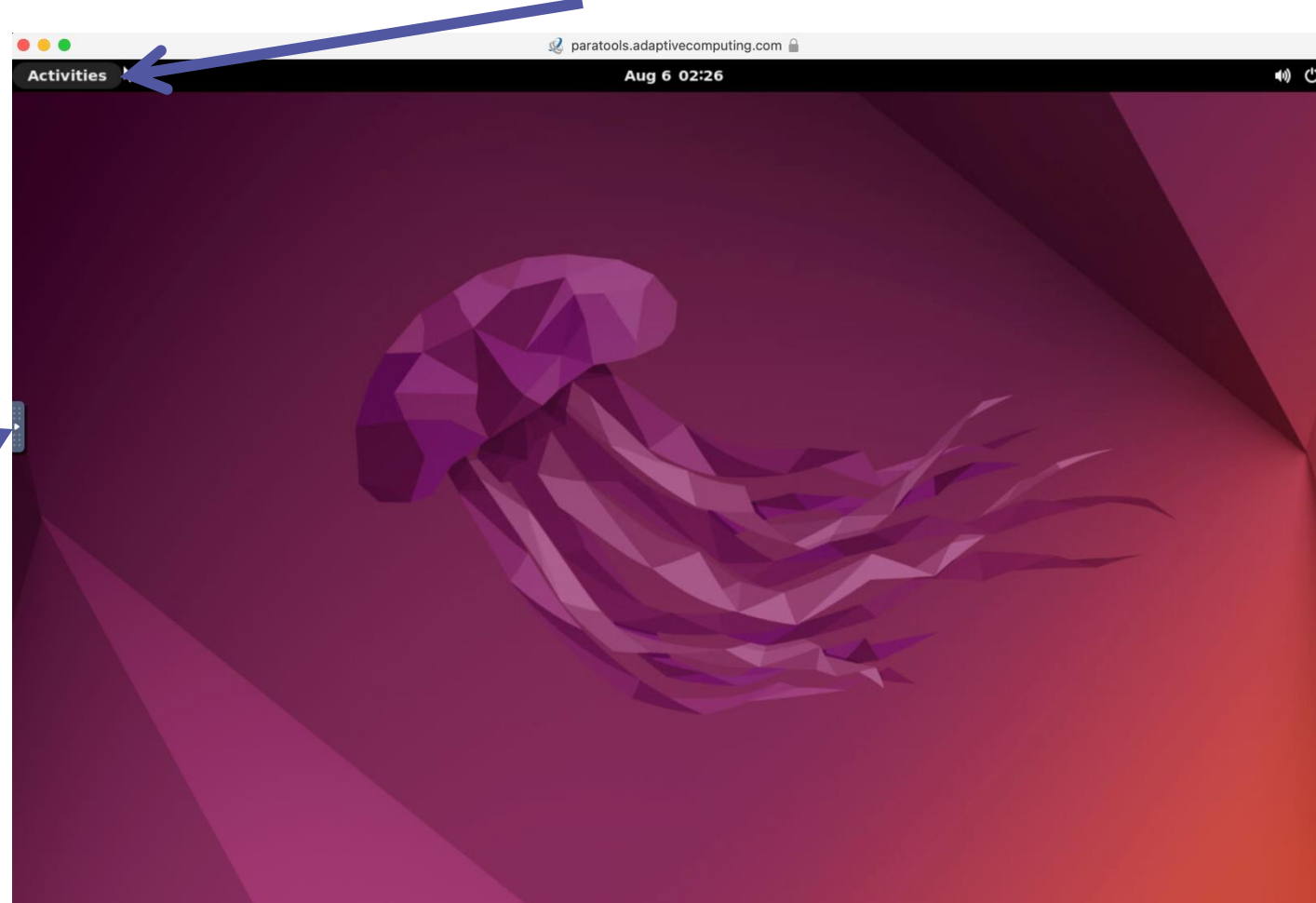


The screenshot shows a web browser window with the address bar displaying paratools.adaptivecomputing.com. A blue arrow points to the address bar. The page features a dark sidebar on the left with navigation links: Heidi - Empowering Minds, Dashboard, COMPUTE / IDE, Heidi Lab, Cluster Manager, CONTENT LIBRARY, Applications, and Courses. The main content area displays the 'Access' tab for a cluster named 'e4s-25-06-aws' on Amazon Web Services. The 'Access' tab includes sections for 'Connection Information' (Public IP Address: 54.70.253.146, Username: training0, Password: 7y19fqvw) and 'Access Methods' (Graphical Access (VNC) with an 'OPEN VNC VIEWER' button, SSH Key Authentication with a 'DOWNLOAD SSH KEY' button, and VPN Access). A 'Connection Instructions' section at the bottom lists steps for SSH connection. The footer indicates 'Copyright © 2025. Adaptive Computing Enterprises, Inc., All rights reserved.'

Connect to Students tab with code 70034 at <https://paratools.adaptivecomputing.com>

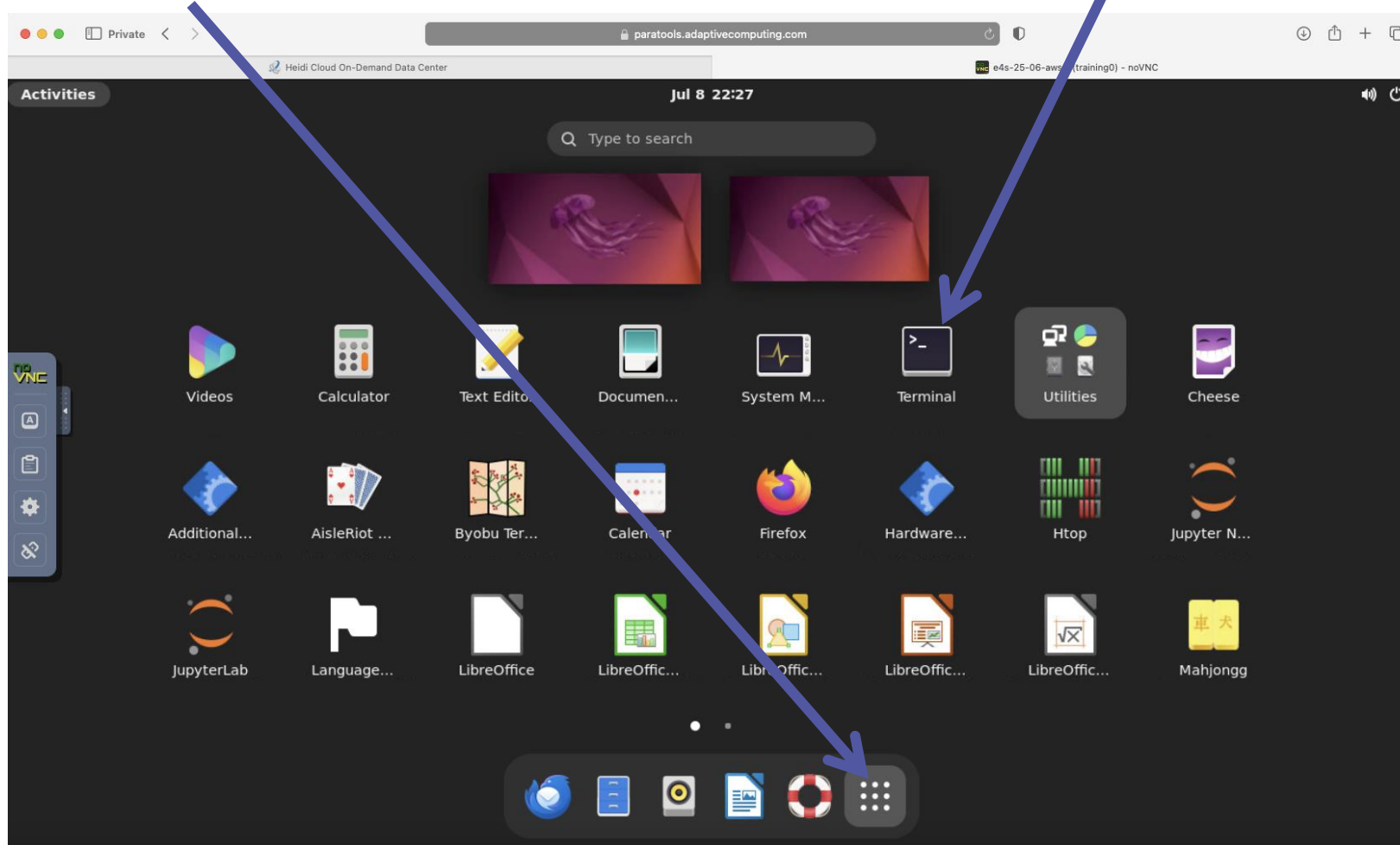
- You should see this jellyfish. Click on Activities.

To copy text from other windows click on the this button to access the clipboard

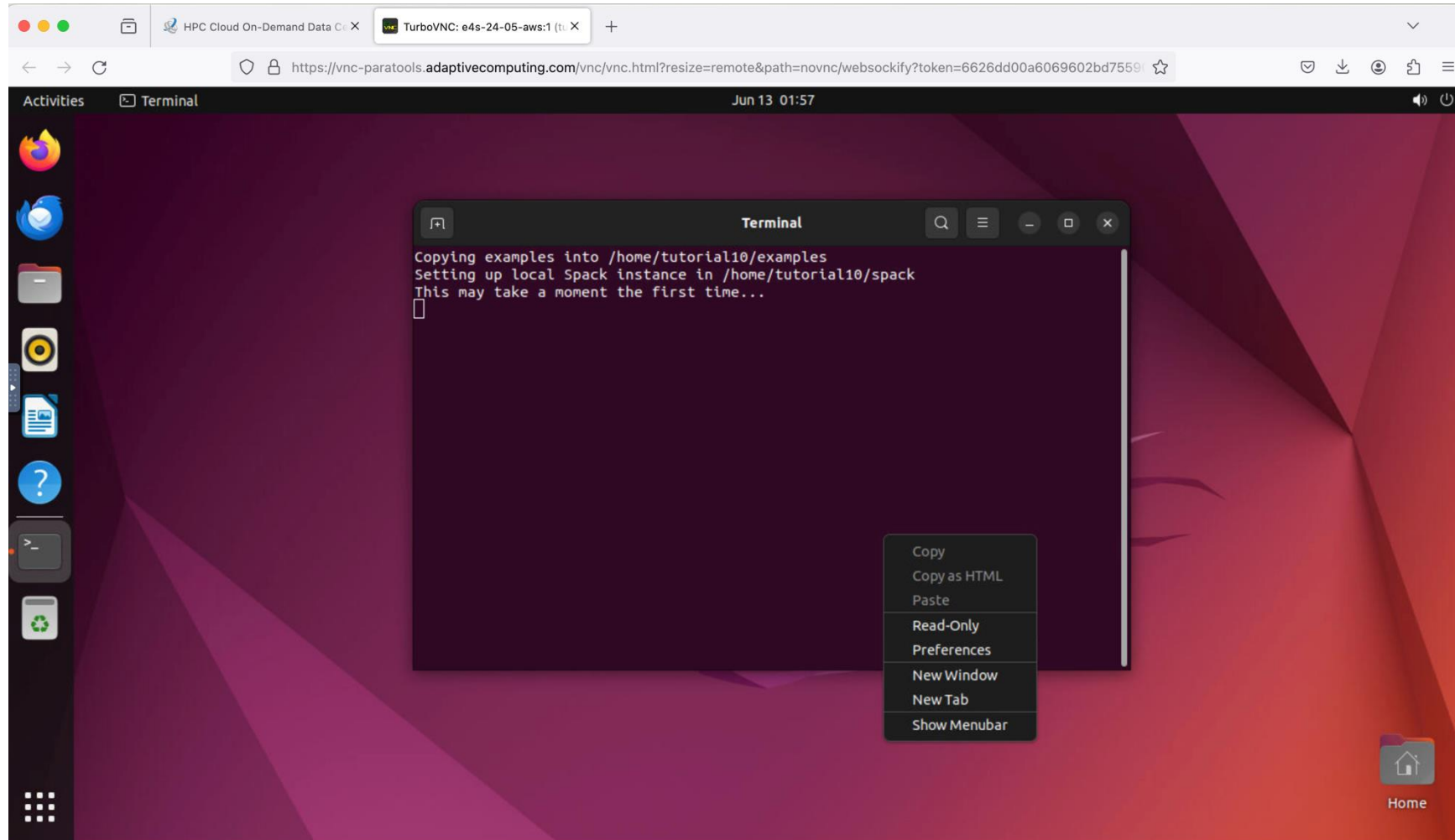


Connect to <https://paratools.adaptivecomputing.com>

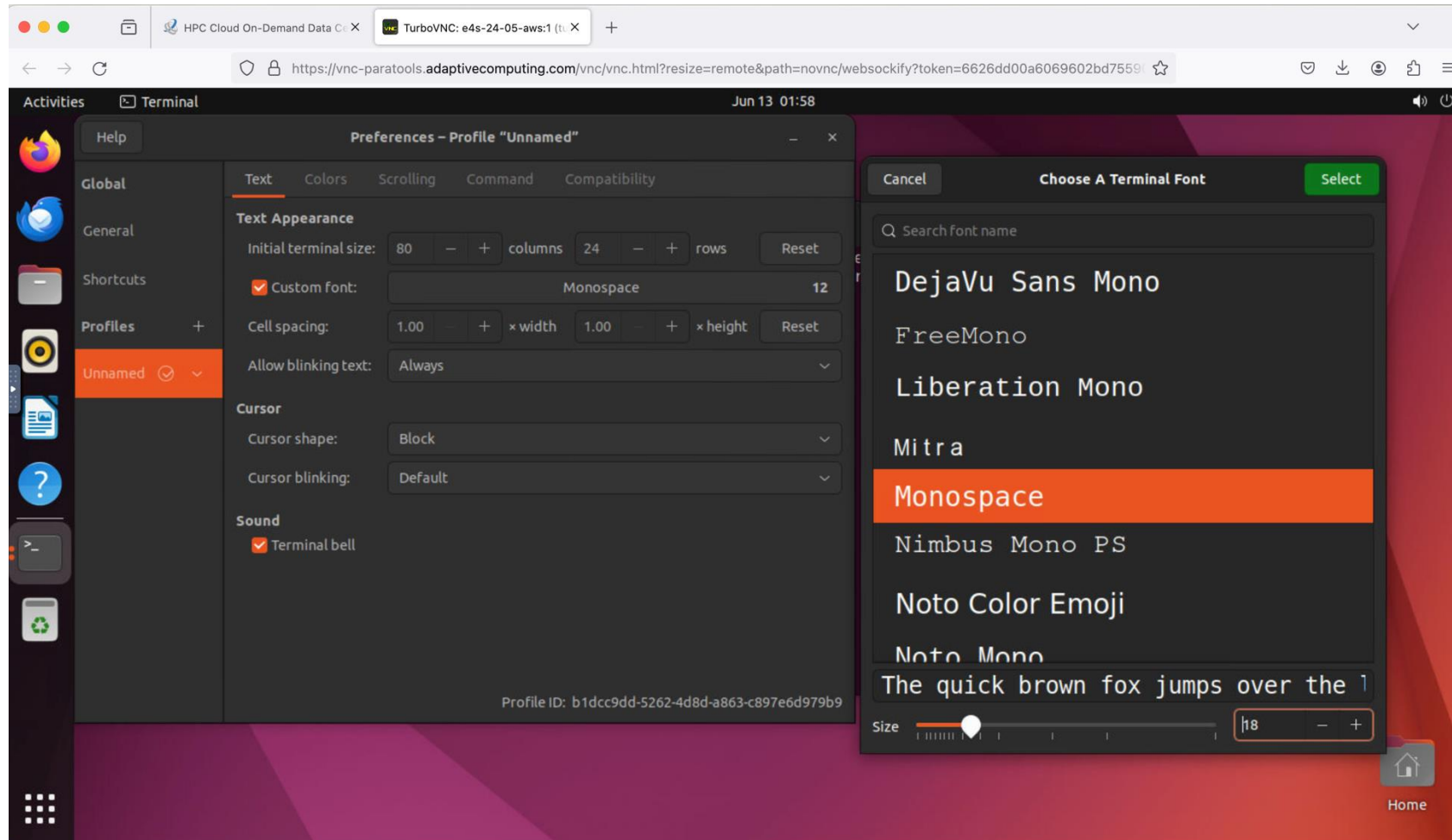
- Click on Activities, nine dots, and then select the Terminal application



To increase font size right click and choose preferences



Choose font size after clicking Custom Font for Terminal

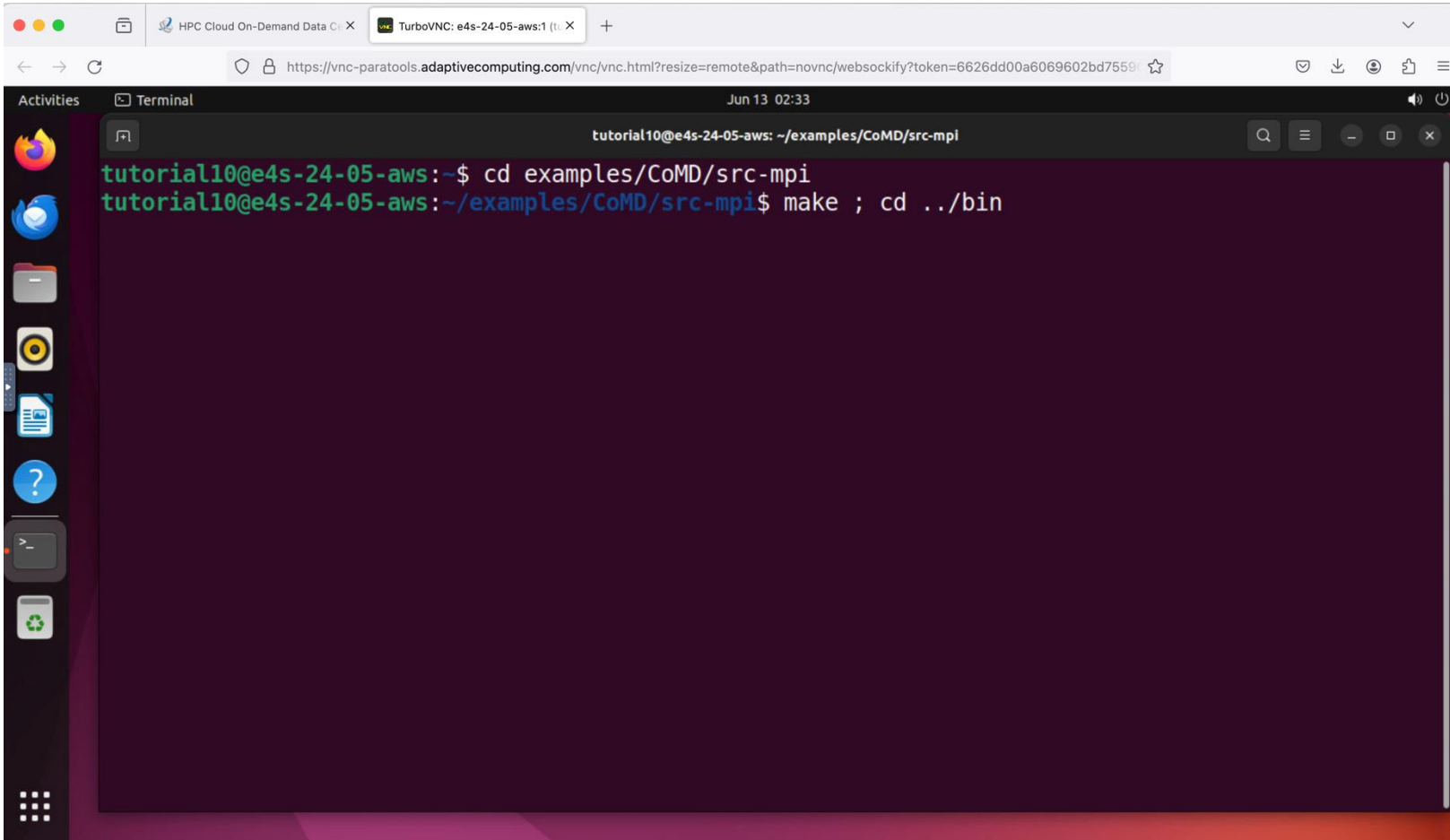


Running your first MPI application on the allocated cluster

```
% cd ~/examples/mpi-procname
% ./compile.sh
% ./run-single-node.sh    # on the login node
% cat mpiprocname.qsub
% qsub mpiprocname.qsub
% qstat -u $USER
% cat mpiprocname.o*

% cd ~/examples/osu-benchmarks
% cat bw.qsub
% qsub bw.qsub
% cat bw.o*              # How close did you get to 50Gbps? At what message size? Multiply MB/s x 8 ...
```

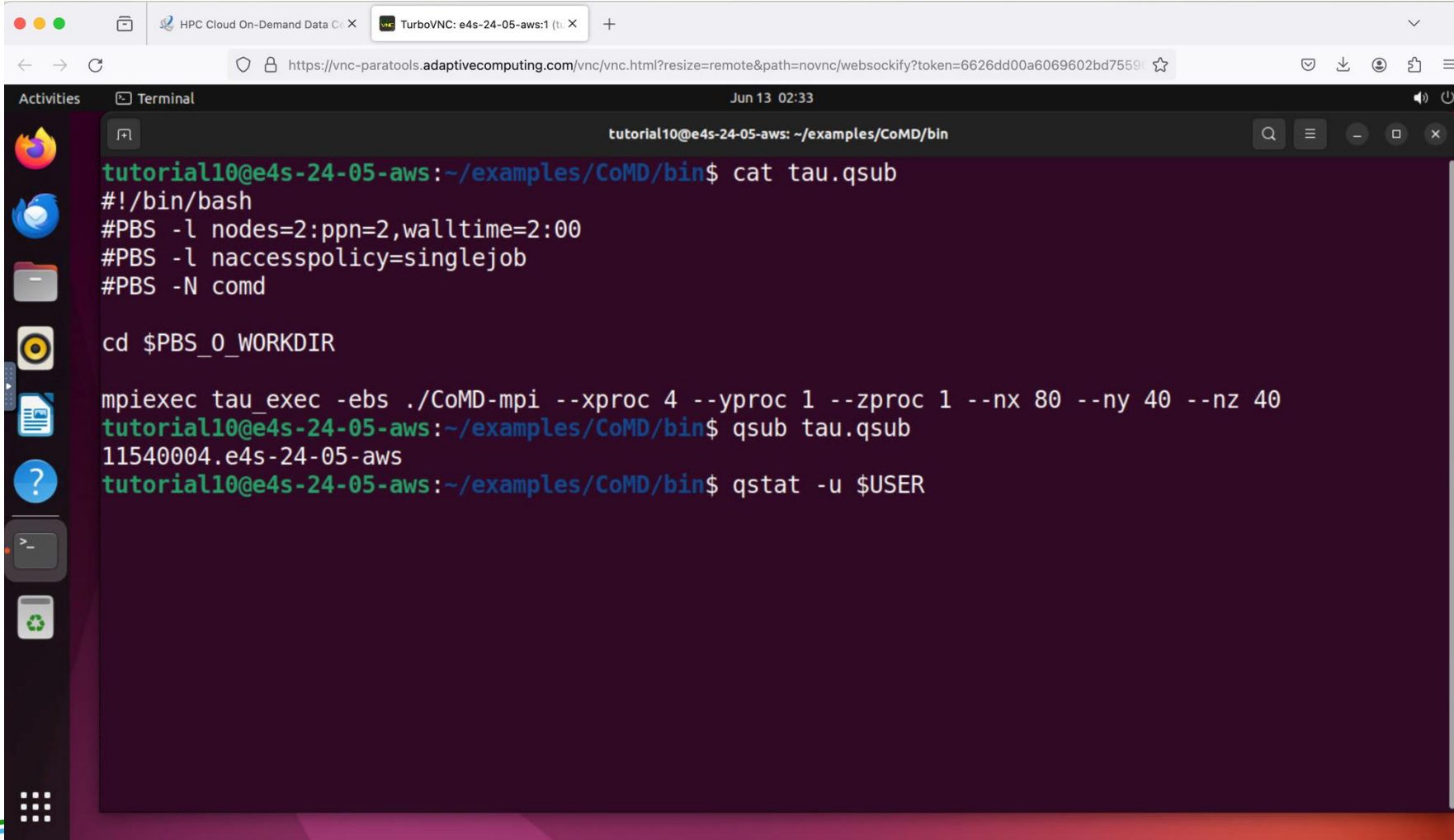
CoMD: TAU with event-based sampling (EBS)



```
tutorial10@e4s-24-05-aws: ~/examples/CoMD/src-mpi
tutorial10@e4s-24-05-aws:~$ cd examples/CoMD/src-mpi
tutorial10@e4s-24-05-aws:~/examples/CoMD/src-mpi$ make ; cd ../bin
```

% cd examples/CoMD/src-mpi
% make; cd ../bin

CoMD: TAU with event-based sampling (EBS)



The screenshot shows a terminal window titled 'tutorial10@e4s-24-05-aws: ~/examples/CoMD/bin'. The terminal output is as follows:

```
tutorial10@e4s-24-05-aws:~/examples/CoMD/bin$ cat tau.qsub
#!/bin/bash
#PBS -l nodes=2:ppn=2,walltime=2:00
#PBS -l naccesspolicy=singlejob
#PBS -N comd

cd $PBS_0_WORKDIR

mpiexec tau_exec -ebs ./CoMD-mpi --xproc 4 --yproc 1 --zproc 1 --nx 80 --ny 40 --nz 40
tutorial10@e4s-24-05-aws:~/examples/CoMD/bin$ qsub tau.qsub
11540004.e4s-24-05-aws
tutorial10@e4s-24-05-aws:~/examples/CoMD/bin$ qstat -u $USER
```

Without TAU:

% qsub comd.qsub

With TAU:

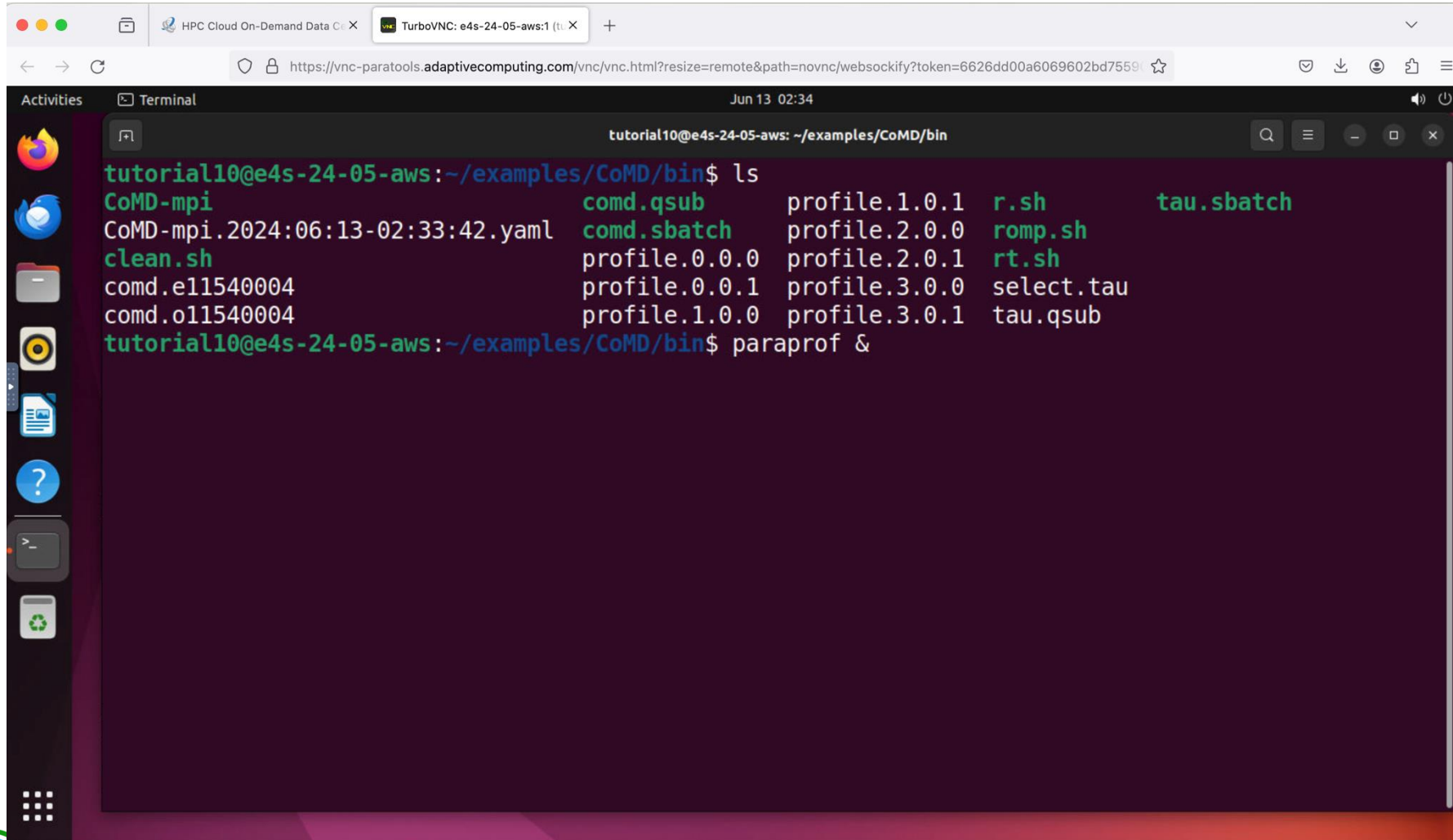
% qsub tau.qsub

% qstat -u \$USER

Also see

../petsc-cuda

CoMD: TAU's paraprof visualizer



The screenshot shows a TurboVNC session with a terminal window titled 'tutorial10@e4s-24-05-aws: ~/examples/CoMD/bin'. The terminal displays the output of the 'ls' command, showing a list of files and directories. The files are arranged in a grid-like fashion, with some files having associated profile names and script names. The terminal prompt is 'tutorial10@e4s-24-05-aws: ~/examples/CoMD/bin\$'.

```
tutorial10@e4s-24-05-aws: ~/examples/CoMD/bin$ ls
CoMD-mpi                               comd.qsub          profile.1.0.1      r.sh              tau.sbatch
CoMD-mpi.2024:06:13-02:33:42.yaml      comd.sbatch       profile.2.0.0      romp.sh
clean.sh                               profile.0.0.0     profile.2.0.1      rt.sh
comd.e11540004                         profile.0.0.1     profile.3.0.0      select.tau
comd.o11540004                         profile.1.0.0     profile.3.0.1      tau.qsub
tutorial10@e4s-24-05-aws: ~/examples/CoMD/bin$ paraprof &
```

% paraprof &

CoMD: TAU's paraprof visualizer

The screenshot shows a web browser window displaying the TAU ParaProf Manager interface. The browser address bar shows the URL: <https://vnc-paratools.adaptivecomputing.com/vnc/vnc.html?resize=remote&path=novnc/websockify?token=6626dd00a6069602bd7559>. The browser window title is "edu-uoregon-tau-paraprof-ParaProf". The main content area shows the TAU ParaProf Manager interface with a sidebar on the left and a main panel on the right. The sidebar contains a tree view of applications, with "Standard Applications" expanded, showing "Default App" and "Default Exp". The "Default Exp" is selected, showing a list of metrics including "TIME". The main panel displays a table of metrics for the selected application, with columns for "Name", "TrialField", and "Value". The table lists various system and application metrics, such as "CPU Cores", "CPU MHz", "CPU Type", "CPU Vendor", "CPUs Allowed", "CPUs Allowed List", "CWD", "Cache Size", "Command Line", "Ending Timestamp", "Executable", "File Type Index", "File Type Name", "Hostname", "Local Time", "MPI Processor Name", "Memories Allowed", "Memories Allowed List", "Memory Size", "Node Name", "OS Machine", "OS Name", "OS Release", "OS Version", and "Starting Timestamp". A context menu is open over the "node 0, thread 0" entry in the table, showing options: "Show Thread Bar Chart", "Show Thread Statistics Text Window", "Show Thread Statistics Table", "Show Thread Call Graph", "Show Thread Event Bar Chart", "Show User Event Bar Chart", "Show User Event Statistics Window", "Show Context Event Window", "Show Metadata for Thread", and "Add Thread to Comparison Window". A blue arrow points from the text "Right click on Node 0, Thread 0 and choose Show Thread Statistics Table (third option)" to the "Show Thread Statistics Table" option in the context menu.

Right click on Node 0, Thread 0 and choose Show Thread Statistics Table (third option)

TAU's ParaProf Profile Browser: Thread Statistics Table

TAU: ParaProf: Statistics for: node 0, thread 0 - /home/tutorial10/examples/CoMD/bin

Name	Exclusive TIME	Inclusive TIME	Calls	Child Calls
.TAU application	0	26.582	1	1
taupreload_main	26.174	26.582	1	1,255
[CONTEXT] taupreload_main	0	25.65	855	0
[SUMMARY] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }]	24.27	24.27	809	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 198 }	4.8	4.8	160	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 209 }	3.78	3.78	126	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 199 }	3.66	3.66	122	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 189 }	2.76	2.76	92	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 202 }	1.98	1.98	56	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 193 }	1.56	1.56	52	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 208 }	1.47	1.47	49	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 207 }	1.38	1.38	46	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 224 }	0.66	0.66	22	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 206 }	0.63	0.63	21	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 223 }	0.39	0.39	13	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 185 }	0.27	0.27	9	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 210 }	0.24	0.24	8	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 220 }	0.18	0.18	6	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 214 }	0.15	0.15	5	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 181 }	0.12	0.12	4	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 175 }	0.09	0.09	3	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 159 }	0.06	0.06	2	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 187 }	0.06	0.06	2	0
[SAMPLE] IjForce [{ /home/tutorial10/examples/CoMD/src-mpi/IjForce.c }] { 156 }	0.03	0.03	1	0
[SUMMARY] getBoxFromCoord [{ /home/tutorial10/examples/CoMD/src-mpi/linkCells.c }]	0.36	0.36	12	0
[SAMPLE] UNRESOLVED /usr/lib/x86_64-linux-gnu/libc.so.6	0.21	0.21	7	0
[SUMMARY] sortAtomsInCell [{ /home/tutorial10/examples/CoMD/src-mpi/haloExchange	0.21	0.21	7	0
[SUMMARY] advancePosition [{ /home/tutorial10/examples/CoMD/src-mpi/advancePosition.c }]	0.12	0.12	4	0

Click on columns to sort (e.g., Inclusive)

Expand nodes and right click on a sample and

Select "Show Source Code"

TAU's ParaProf Profile Browser: Source Code Browser

The screenshot displays the TAU's ParaProf Profile Browser interface. On the left, a source code editor shows the file `/home/tutorial10/examples/CoMD/src-mpi/ljForce.c`. The code is a C program for molecular dynamics simulation, featuring nested loops for iterating over boxes and atoms. Line 198 is highlighted, showing a calculation of the distance between two atoms.

On the right, a performance table is displayed, showing the execution time and call counts for various code blocks. The table has four columns: Exclusive TIME, Inclusive TIME, Calls, and Child Calls. The rows represent different levels of the code hierarchy, with the top row showing the total execution time of 26.582 seconds.

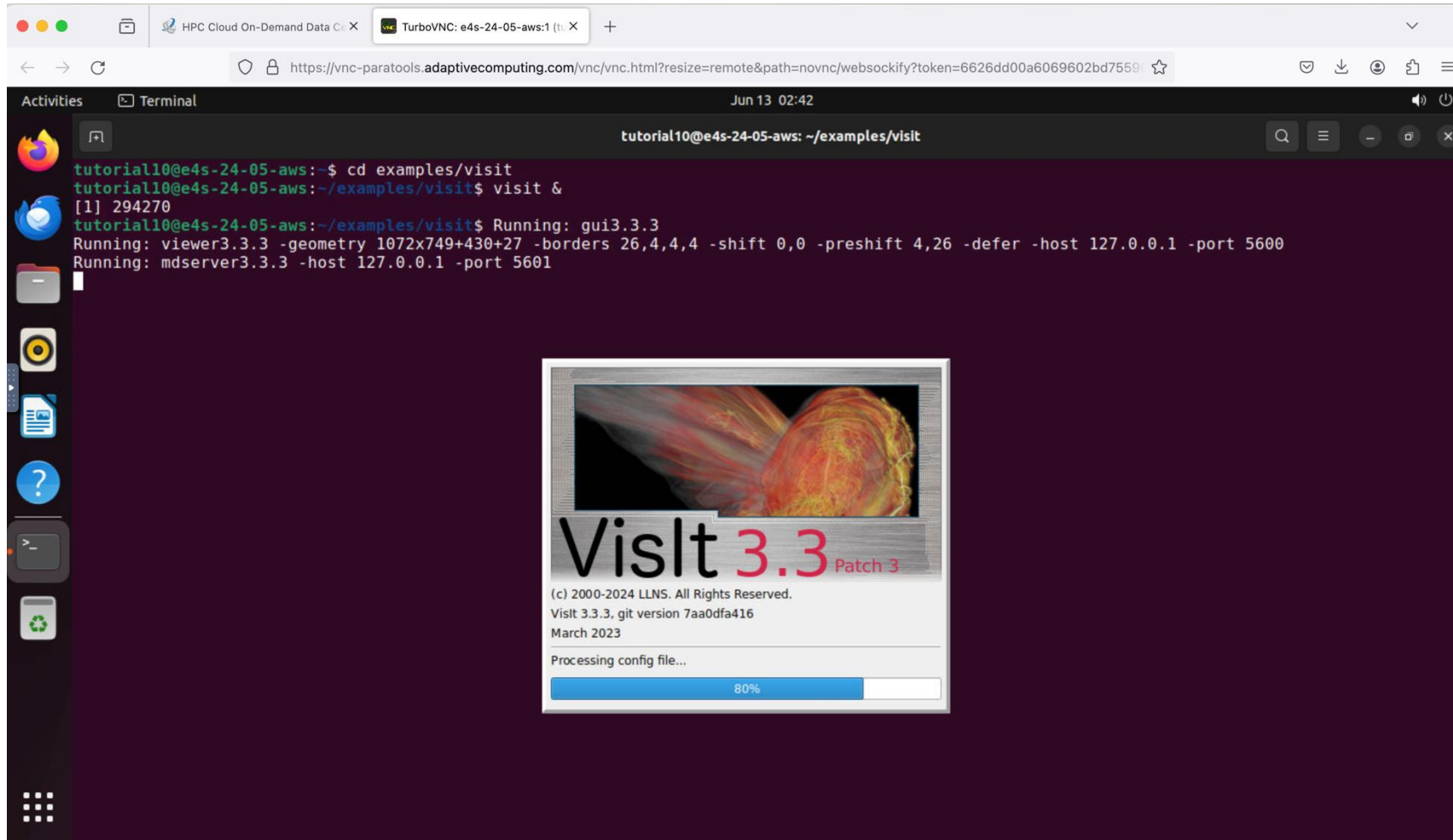
Exclusive TIME	Inclusive TIME	Calls	Child Calls
0	26.582	1	1
26.174	26.582	1	1,255
0	25.65	855	0
24.27	24.27	809	0
4.8	4.8	160	0
3.78	3.78	126	0
3.66	3.66	122	0
2.76	2.76	92	0
1.98	1.98	66	0
1.56	1.56	52	0
1.47	1.47	49	0
1.38	1.38	46	0
0.66	0.66	22	0
0.63	0.63	21	0
0.39	0.39	13	0
0.27	0.27	9	0
0.24	0.24	8	0
0.18	0.18	6	0
0.15	0.15	5	0
0.12	0.12	4	0
0.09	0.09	3	0
0.06	0.06	2	0
0.06	0.06	2	0
0.03	0.03	1	0
0.36	0.36	12	0
0.21	0.21	7	0
0.21	0.21	7	0
0.12	0.12	4	0

The application spent 4.8 seconds at line 198 in `ljForce.c` in MPI rank 0. TAU collected 160 samples at this line of code.

It is within five levels of for loops!

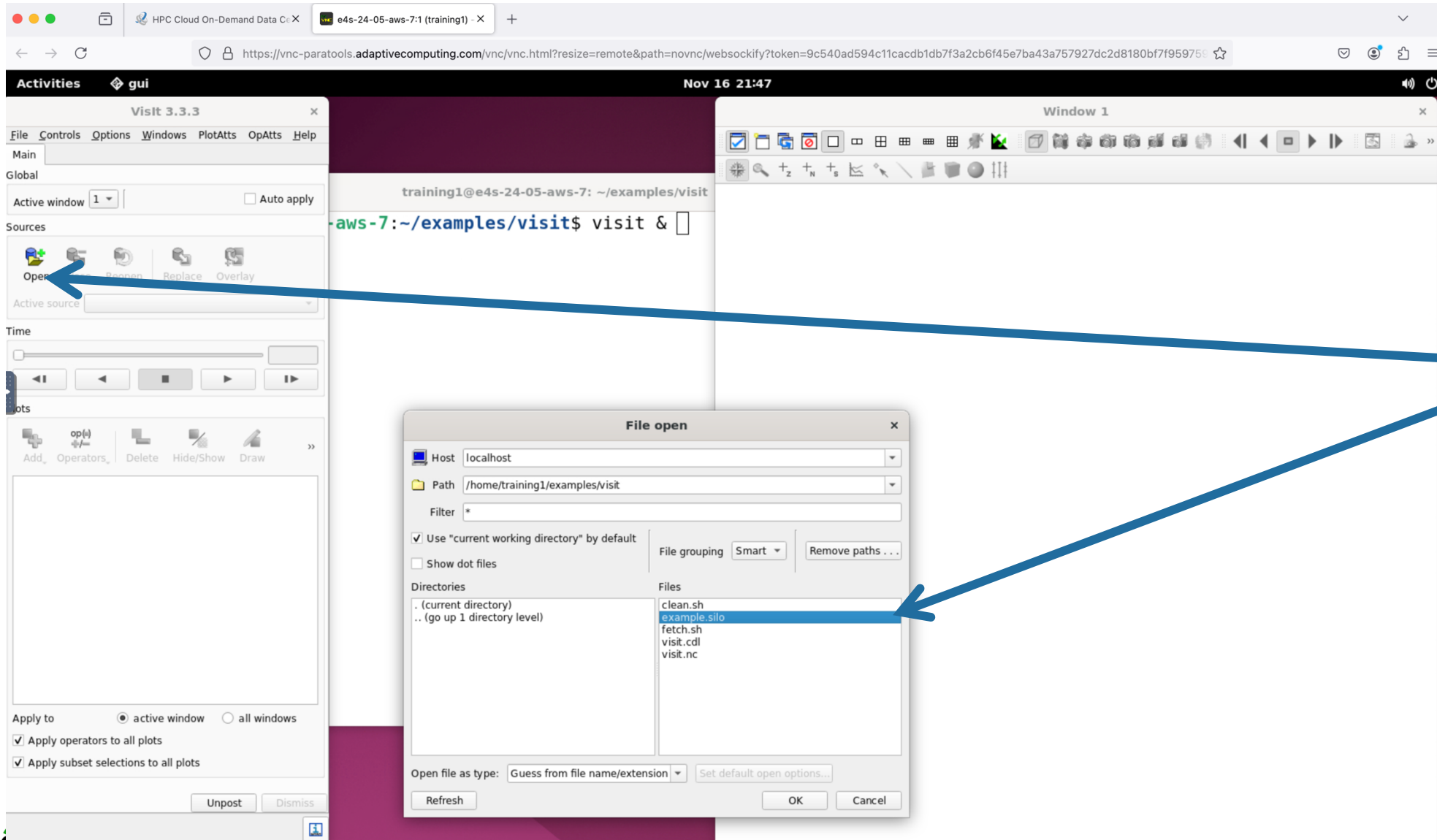
There was no change to source code, build system, or the application binary!

VisIt visualizer: 3D graphics on remote desktop



cd ~/workshop/visit
visit &

VisIt visualizer: 3D graphics on remote desktop

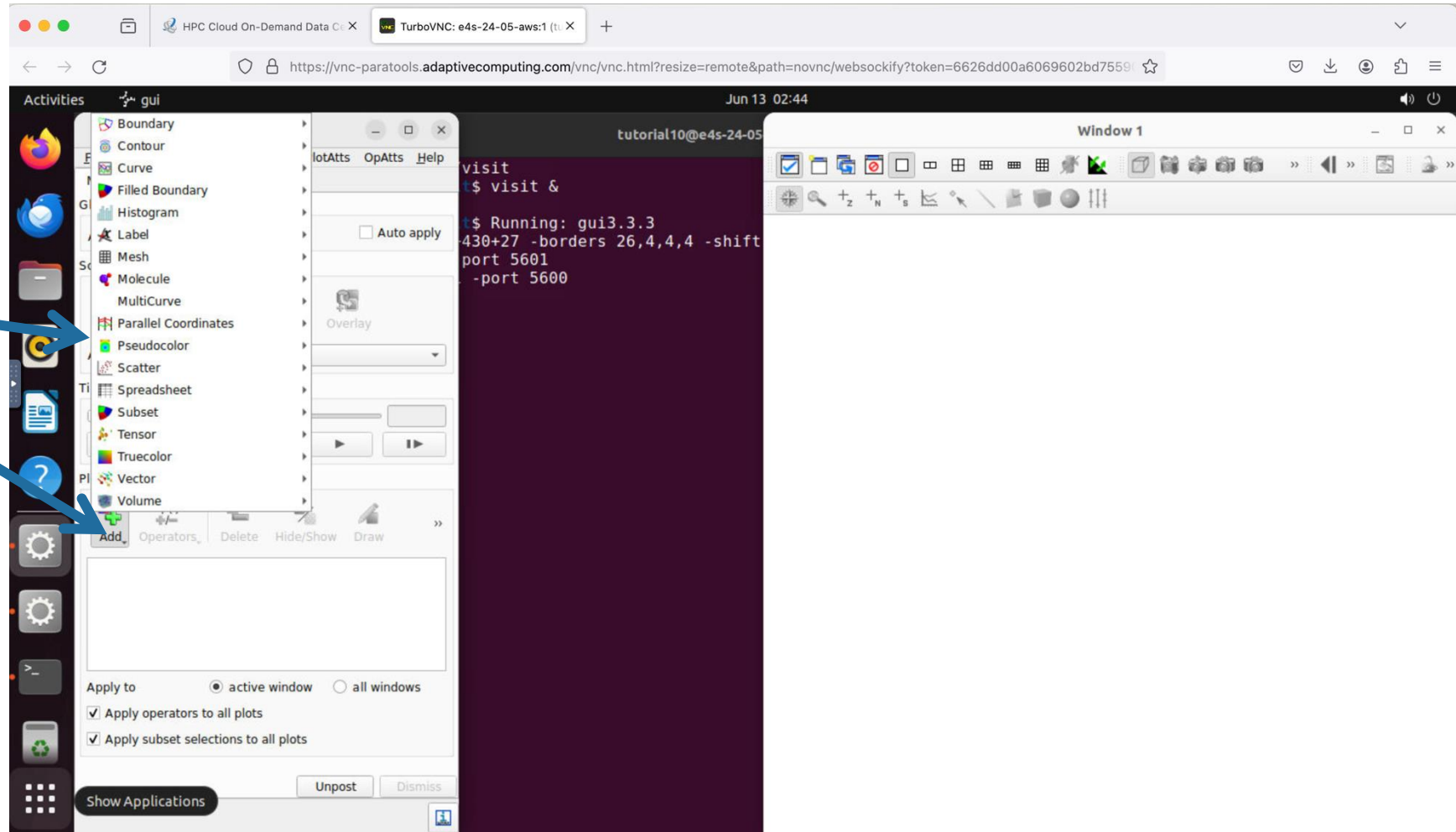


cd workshop/visit
visit &

Open example.silo

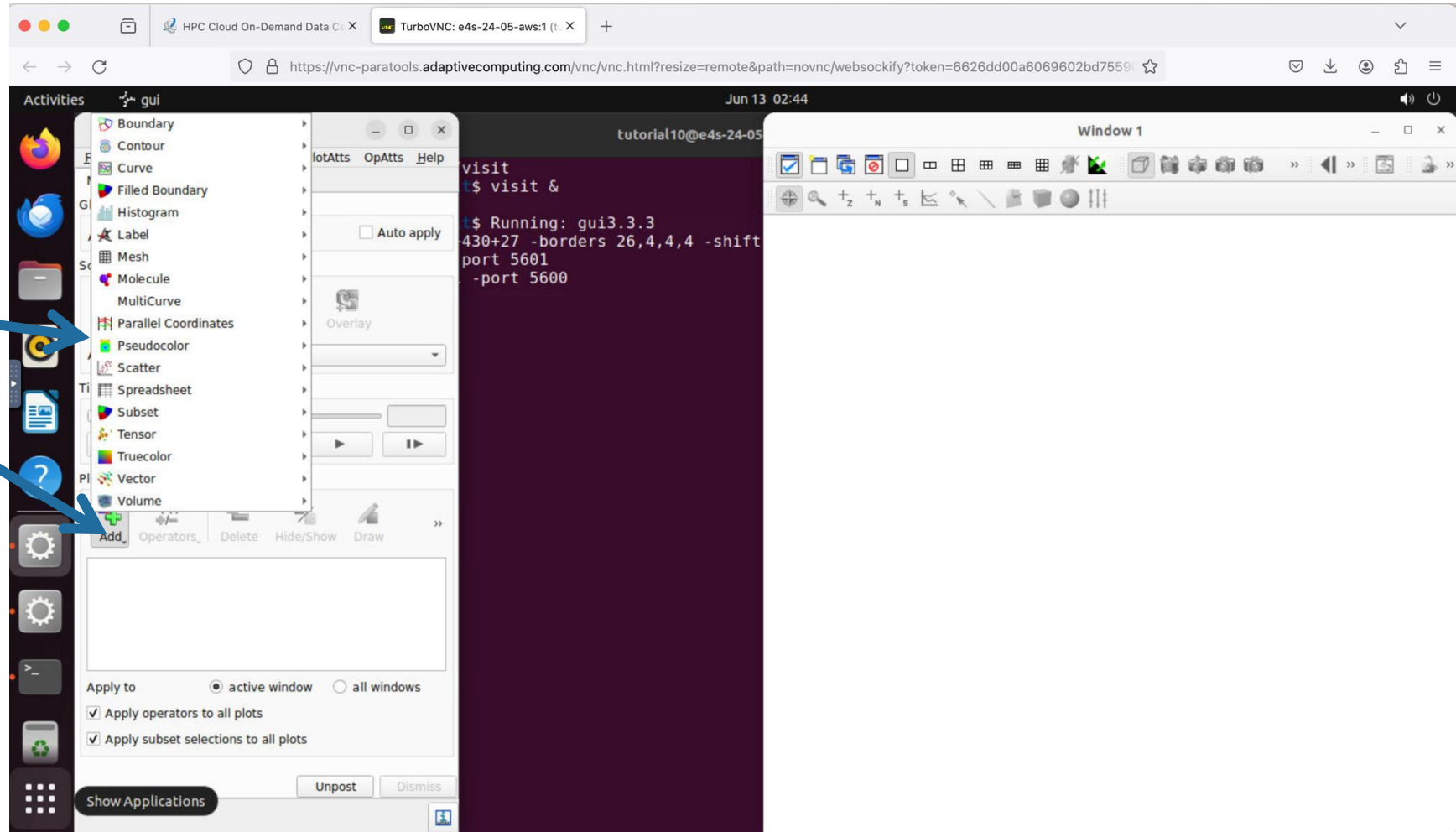
VisIt visualizer

Click Add
Add Pseudocolor ->
Pressure



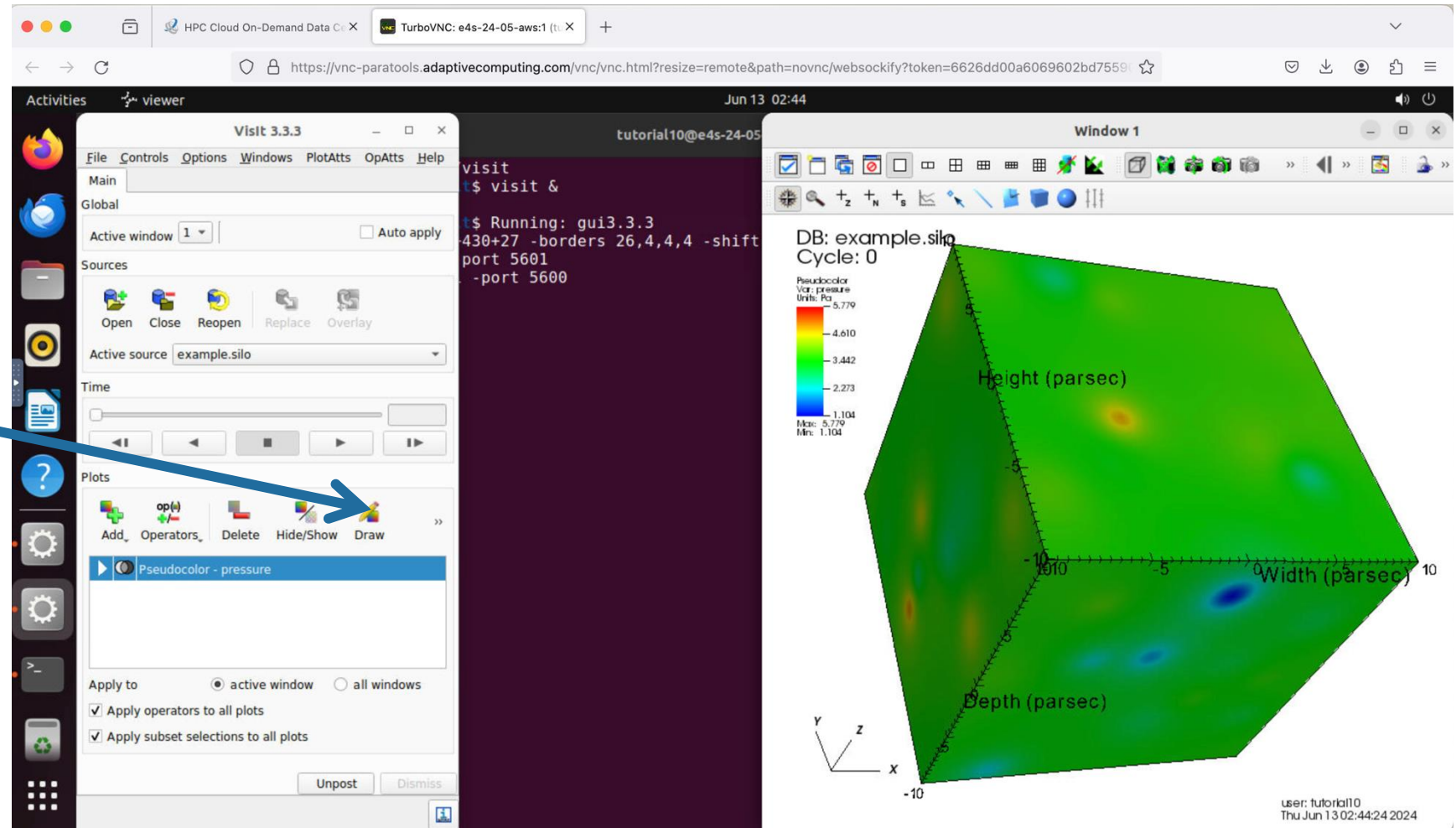
VisIt visualizer

Click Add
Add Pseudocolor ->
Pressure
Click Draw



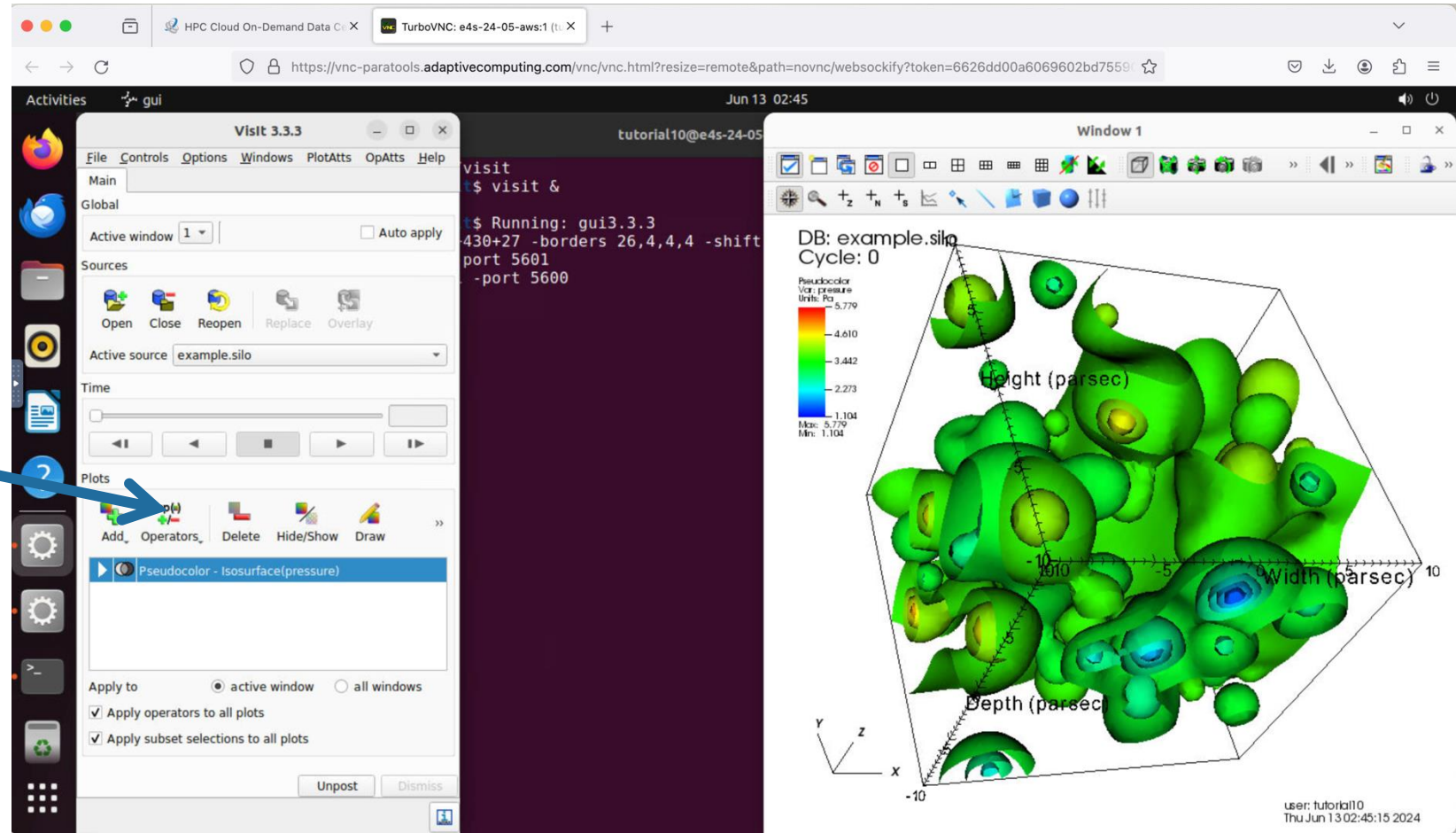
VisIt visualizer

Add Pseudocolor ->
Pressure
Click Draw
Rotate image

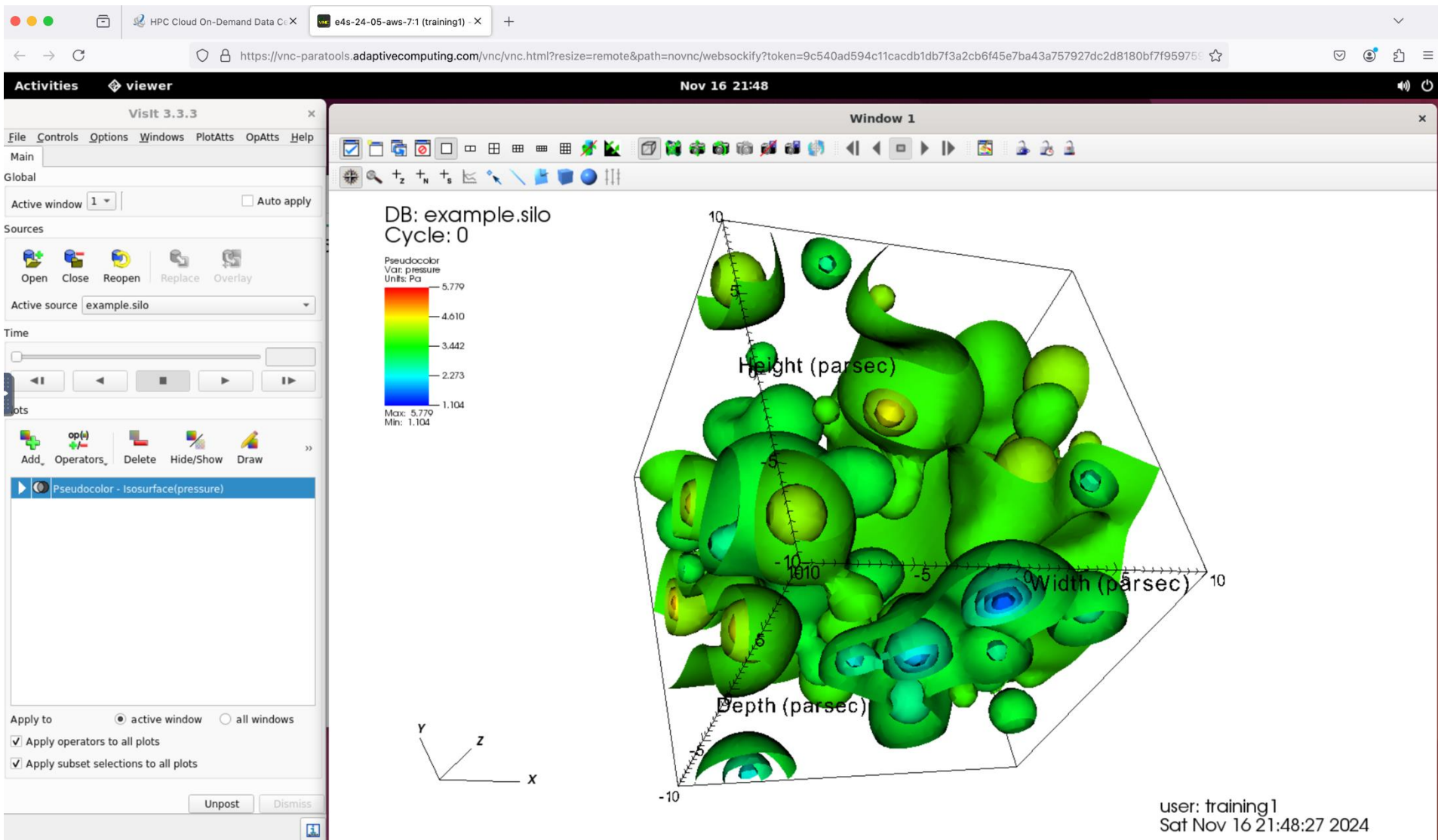


VisIt visualizer

Add Operators ->
Isosurface
Click Draw
Rotate image



VisIt visualizer: 3D graphics check



Other tools to try

```
% cd ~/examples/visit
```

```
% visit &
```

Open example.silo -> Add -> Pseudocolor -> Pressure -> Draw

Options -> Slicing -> Isosurface -> Draw

```
% paraview &
```

Example Visualizations -> Pick one.

```
% python
```

```
>>> import nemo
```

```
>>> import <your_favorite_package>
```

If it is not installed:

```
pip install <package>
```

Other tools to try

% which adk
Google Agent Development Kit

% cd ~/workshop/pytorch

% codium &

Install Python extensions -> Open folder ~/workshop/pytorch. Open f2.py click run

% which npm

If you have a Claude paid subscription from Anthropic:

% npm install -g @anthropic-ai/claude-code

% which claude

% claude

...

% jupyter notebook &

% marimo new

AI packages included in ParaTools Pro for E4S™

- PyTorch
- TensorFlow
- NVIDIA NeMo™
- NVIDIA BioNeMo™
- VLLM
- Pandas, Scikit-Learn, OpenCV, Jax, Horovod
- Google Agent Development Kit (ADK)
- Support for installation of third party tools (e.g., Anthropic's Claude code using npm, Node-JS)
- Jupyter Notebook, Codium, and Marimo reactive notebooks
- Commonly used Python tools and libraries with support for NVIDIA GPUs on the 4 major CSPs

Acknowledgment

- This work was supported by the U.S. Department of Energy, Office of Science, Advanced Computing Research, through the Next-Generation Scientific Software Technologies (NGSST) under contract DE-AC02-AC05-00OR22725, DOE SBIR DE-SC0022502, and NNSA Sandia contract 2542488.*



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.

