E4S: Extreme-scale Scientific Software Stack

ECP Annual Meeting 2021 Tutorial
https://www.ecpannualmeeting.com

Monday, April 12, 2021, 7:00am – 10am PT (with a 10 min break)

Sameer Shende and Luke Peyralans, U. Oregon
David H. Rogers, LANL
Jon Rood, NREL

https://e4s.io/talks/E4S_Tut_ECP_AM21.pdf

Acknowledgments

Performance Research Laboratory OACISS, University of Oregon

- Luke Peyralans
- Wyatt Spear
- Camille Coti
- Erik Keever
- Jean-Baptiste Skutnik
- Frederick Deny
- Robert Yelle

- Andrew J. Younge, Sandia National Laboratories
- Shane Canon, Berkeley Lab
- Todd Gamblin, LLNL
- Tamara Dahlgren, LLNL
- Shahzeb Siddiqui, Berkeley Lab
- Mike Heroux, Sandia National Laboratories
- Jim Willenbring, Sandia National Laboratories
- Chuck Atkins, Kitware
- Tim Haines, University of Wisconsin, Madison
- Ulreke Meier Yang, LLNL
- Todd Munson, ANL

Challenges

- As our software gets more complex, it is getting harder to install tools and libraries correctly in an integrated and interoperable software stack.

E4S: Better Quality, Documentation, Test, Integration, Delivery, Build & Use

Delivering HPC software to facilities, vendors, agencies, industry, international partners in a brand-new way

Community Policies
Commitment to software quality

DocPortal
Single portal to all E4S product info

Portfolio testing
Especially leadership platforms

Curated collection
The end of dependency hell

Quarterly releases
Release 1.2 – November

Build caches
10X build time improvement

Turnkey stack
A new user experience

https://e4s.io

E4S Strategy Group
US agencies, industry, international

EXASCALE COMPUTING PROJECT

E4S: Extreme-scale Scientific Software Stack

- Curated, Spack based software distribution
- Spack binary build caches for bare-metal installs
  - x86_64, ppc64le (IBM Power 9), and aarch64 (ARM64)
- Container images on DockerHub and E4S website of pre-built binaries of ECP ST products
- Base images and full featured containers (with GPU support)
- GitHub recipes for creating custom images from base images
- GitLab integration for building E4S images
- E4S validation test suite on GitHub
- E4S-cl container launcher tool for MPI substitution in applications using MPICH ABI
- E4S VirtualBox image with support for container runtimes
  - Docker
  - Singularity
  - Shifter
  - Charliecloud
- AWS and GCP images to deploy E4S

https://e4s.io

E4S Tutorial Goals

- Reserve an AWS instance by editing the Google doc at https://e4s.io/tutorial
- Use DCV [AWS] to login to https://tut<XXX>.supercontainers.org:8443/#e4s
  - Login: tutorial
  - Password: HPCLinux12!
- Hello numerical world! BSSw tutorial example (VisIT for visualization)
- Introduction to containers for HPC (Docker, Singularity, Shifter, MPI builds)
- Run NPB LU and Trilinos (Zoltan) and use E4S-cl to replace MPICH (container) with Intel MPI
- Pantheon demo (Ascent, Cinema visualization in Jupyter notebook)
- Nalu-Wind demo (Spack build cache usage, Kokkos, TAU Performance System®, Paraview)
- Container images on DockerHub and E4S website of pre-built binaries of ECP ST products
- AI/ML integration in E4S with Intel oneAPI demo with PyTorch and Tensorflow
- Pointers to other tutorials at ECP AM:
  - Using Spack to Accelerate Developer Workflows (Fri, Apr. 16 10AM – 6PM ET)
  - Intro to Containers for HPC (Wednesday, April 14th, 1:00 PM – 2:30 PM ET)

ECP's three technical areas have the necessary components to meet national goals

Performant mission and science applications @ scale

Aggressive RD&D Project	Mission apps & integrated S/W stack	Deployment to DOE HPC Facilities	Hardware tech advances

Application Development (AD)

Develop and enhance the predictive capability of applications critical to the DOE

24 applications including national security, to energy, earth systems, economic security, materials, and data

Software Technology (ST)

Deliver expanded and vertically integrated software stack to achieve full potential of exascale computing

70 unique software products spanning programming models and run times, math libraries, data and visualization

Hardware and Integration (HI)

Integrated delivery of ECP products on targeted systems at leading DOE HPC facilities

6 US HPC vendors focused on exascale node and system design; application integration and software deployment to facilities

We work on products applications need now and into the future

Key themes:
- Exploration/development of new algorithms/software for emerging HPC capabilities:
- High-concurrency node architectures and advanced memory & storage technologies.
- Enabling access and use via standard APIs.

Software categories:
- The next generation of well-known and widely used HPC products (e.g., MPICH, OpenMPI, PETSc)
- Some lesser used but known products that address key new requirements (e.g., Kokkos, RAJA, Spack)
- New products that enable exploration of emerging HPC requirements (e.g., SICM, zfp, UnifyCR)

Example Products	Engagement
MPI – Backbone of HPC apps	Explore/develop MPICH and OpenMPI new features & standards.
OpenMP/OpenACC –On-node parallelism	Explore/develop new features and standards.
Performance Portability Libraries	Lightweight APIs for compile-time polymorphisms.
LLVM/Vendor compilers	Injecting HPC features, testing/feedback to vendors.
Perf Tools - PAPI, TAU, HPCToolkit	Explore/develop new features.
Math Libraries: BLAS, sparse solvers, etc.	Scalable algorithms and software, critical enabling technologies.
IO: HDF5, MPI-IO, ADIOS	Standard and next-gen IO, leveraging non-volatile storage.
Viz/Data Analysis	ParaView-related product development, node concurrency.

Software Technology Ecosystem

Levels of Integration | Product | Source and Delivery

ECP ST Individual Products

- Standard workflow
- Existed before ECP

ST Products

Source: ECP L4 teams; Non-ECP Developers; Standards Groups
Delivery: Apps directly; spack; vendor stack; facility stack

- Group similar products
- Make interoperable
- Assure policy compliant
- Include external products

SDKs

Source: ECP SDK teams; Non-ECP Products (policy compliant, spackified)
Delivery: Apps directly; spack install sdk; future: vendor/facility

- Build all SDKs
- Build complete stack
- Containerize binaries

E4S

Source: ECP E4S team; Non-ECP Products (all dependencies)
Delivery: spack install e4s; containers; CI Testing

ECP ST Open Product Integration Architecture

E4S Community Policies V1.0 Released

What is E4S?

The Extreme-scale Scientific Software Stack (E4S) is a community effort to provide open source software packages for developing, deploying and running scientific applications on high-performance computing (HPC) platforms. E4S provides from-source builds and containers of a broad collection of HPC software packages.

Purpose

E4S exists to accelerate the development, deployment and use of HPC software, lowering the barriers for HPC users. E4S provides containers and turn-key, from-source builds of more than 80 popular HPC products in programming models, such as MPI; development tools such as HPCToolkit, TAU and PAPI; math libraries such as PETSc and Trilinos; and Data and Viz tools such as HDF5 and Paraview.

Approach

By using Spack as the meta-build tool and providing containers of pre-built binaries for Docker, Singularity, Shifter and CharlieCloud, E4S enables the flexible use and testing of a large collection of reusable HPC software packages.

E4S Community Policies Version 1

A Commitment to Quality Improvement

- Will serve as membership criteria for E4S
  - Membership is not required for *inclusion* in E4S
  - Also includes forward-looking draft policies
- Purpose: enhance sustainability and interoperability
- Topics cover building, testing, documentation, accessibility, error handling and more
- Multi-year effort led by SDK team
  - Included representation from across ST
  - Multiple rounds of feedback incorporated from ST leadership and membership
- Modeled after xSDK Community Policies
- https://e4s-project.github.io/policies.html

P1 *Spack-based Build and Installation* Each E4S member package supports a scriptable *Spack* build and production-quality installation in a way that is compatible with other E4S member packages in the same environment. When E4S build, test, or installation issues arise, there is an expectation that teams will collaboratively resolve those issues.

P2 *Minimal Validation Testing* Each E4S member package has at least one test that is executable through the E4S validation test suite (*https://github.com/E4S-Project/testsuite*). This will be a post-installation test that validates the usability of the package. The E4S validation test suite provides basic confidence that a user can compile, install and run every E4S member package. The E4S team can actively participate in the addition of new packages to the suite upon request.

P3 *Sustainability* All E4S compatibility changes will be sustainable in that the changes go into the regular development and release versions of the package and should not be in a private release/branch that is provided only for E4S releases.

P4 *Documentation* Each E4S member package should have sufficient documentation to support installation and use.

P5 *Product Metadata* Each E4S member package team will provide key product information via metadata that is organized in the *E4S DocPortal* format. Depending on the filenames where the metadata is located, this may require *minimal setup*.

P6 *Public Repository* Each E4S member package will have a public repository, for example at GitHub or Bitbucket, where the development version of the package is available and pull requests can be submitted.

P7 *Imported Software* If an E4S member package imports software that is externally developed and maintained, then it must allow installing, building, and linking against a functionally equivalent outside copy of that software. Acceptable ways to accomplish this include (1) forsaking the internal copied version and using an externally-provided implementation or (2) changing the file names and namespaces of all global symbols to allow the internal copy and the external copy to coexist in the same downstream libraries and programs. This pertains primarily to third party support libraries and does not apply to key components of the package that may be independent packages but are also integral components to the package itself.

P8 *Error Handling* Each E4S member package will adopt and document a consistent system for signifying error conditions as appropriate for the language and application. For e.g., returning an error condition or throwing an exception. In the case of a command line tool, it should return a sensible exit status on success/failure, so the package can be safely run from within a script.

P9 *Test Suite* Each E4S member package will provide a test suite that does not require special system privileges or the purchase of commercial software. This test suite should grow in its comprehensiveness over time. That is, new and modified features should be included in the suite.

EXASCALE COMPUTING PROJECT

Extreme-scale Scientific Software Stack (E4S)

- E4S: A Spack-based distribution of ECP ST and related and dependent software tested for interoperability and portability to multiple architectures
- Provides distinction between SDK usability / general quality / community and deployment / testing goals
- Will leverage and enhance SDK interoperability thrust
- Oct 2018: E4S 0.1 - 24 full, 24 partial release products
- Jan 2019: E4S 0.2 - 37 full, 10 partial release products
- Nov 2019: E4S 1.0 - 50 full, 5 partial release products
- Jan 2020: E4S 1.1 – ppc64le and x86_64 release with 50 full (x86_64), 46 full (ppc64le) release products.
- Nov. 2020: E4S 1.2 – ppc64le and x86_64 release each with 67 full release products with support for ROCm, and CUDA.
- Feb 2021: E4S support for Intel OneAPI, llvm-doe (Clang 13)

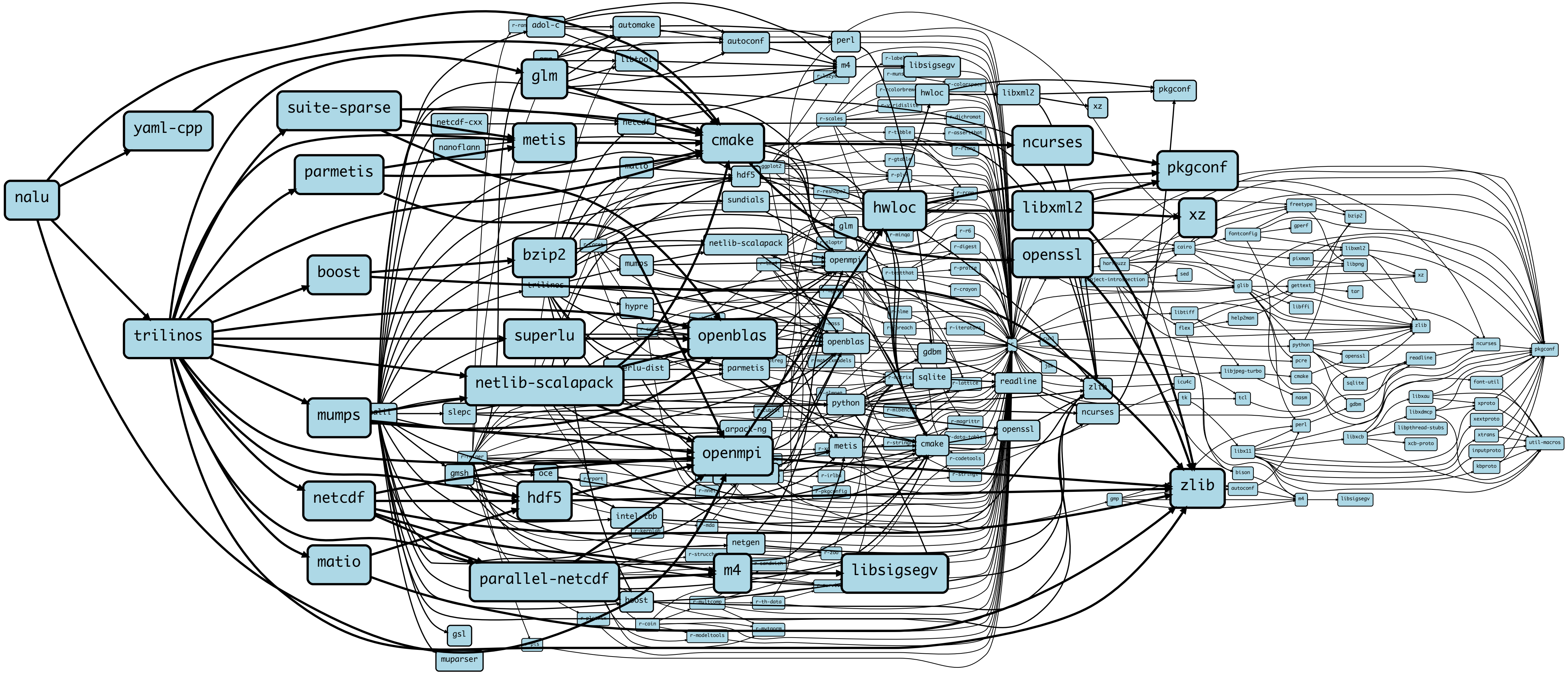
E4S

https://e4s.io

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

Scientific software is becoming extremely complex

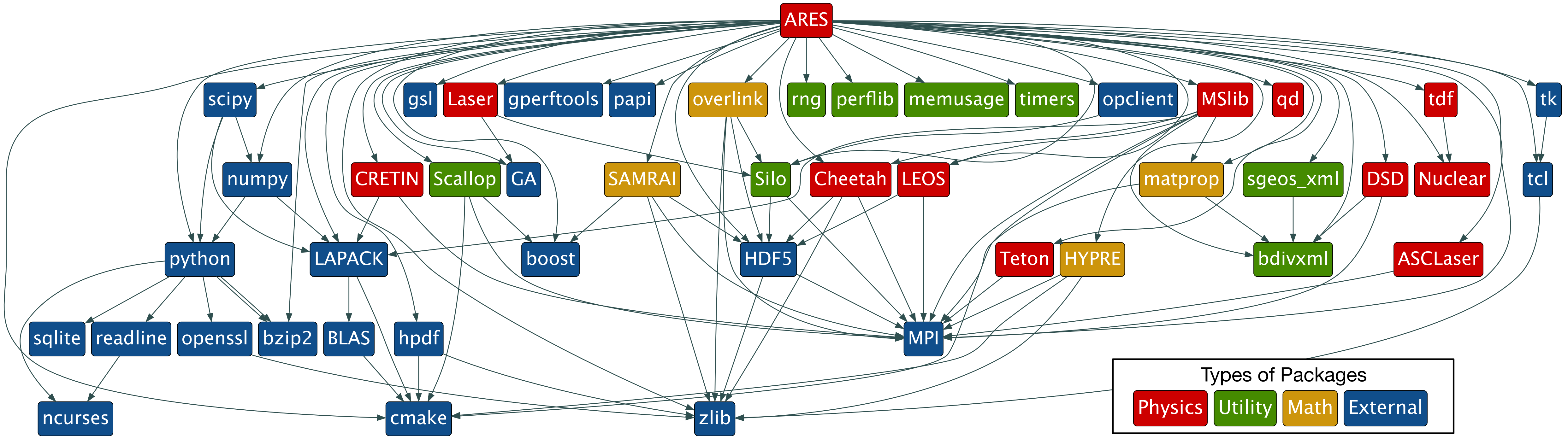


Nalu: Generalized Unstructured Massively Parallel Low Mach Flow

dealii: C++ Finite Element Library

R Miner: R Data Mining Library

Even proprietary codes are based on many open source libraries



- Half of this DAG is external (blue); *more* than half of it is open source
- Nearly *all* of it needs to be built specially for HPC to get the best performance

The Exascale Computing Project is building an entire *ecosystem*

15+ applications x **80+ software packages** x **5+ target architectures/platforms**
Xeon Power KNL
NVIDIA ARM Laptops?

x **Up to 7 compilers**
Intel GCC Clang XL
PGI Cray NAG

x **10+ Programming Models**
OpenMPI MPICH MVAPICH OpenMP CUDA
OpenACC Dharma Legion RAJA Kokkos

x **2-3 versions of each package + external dependencies**

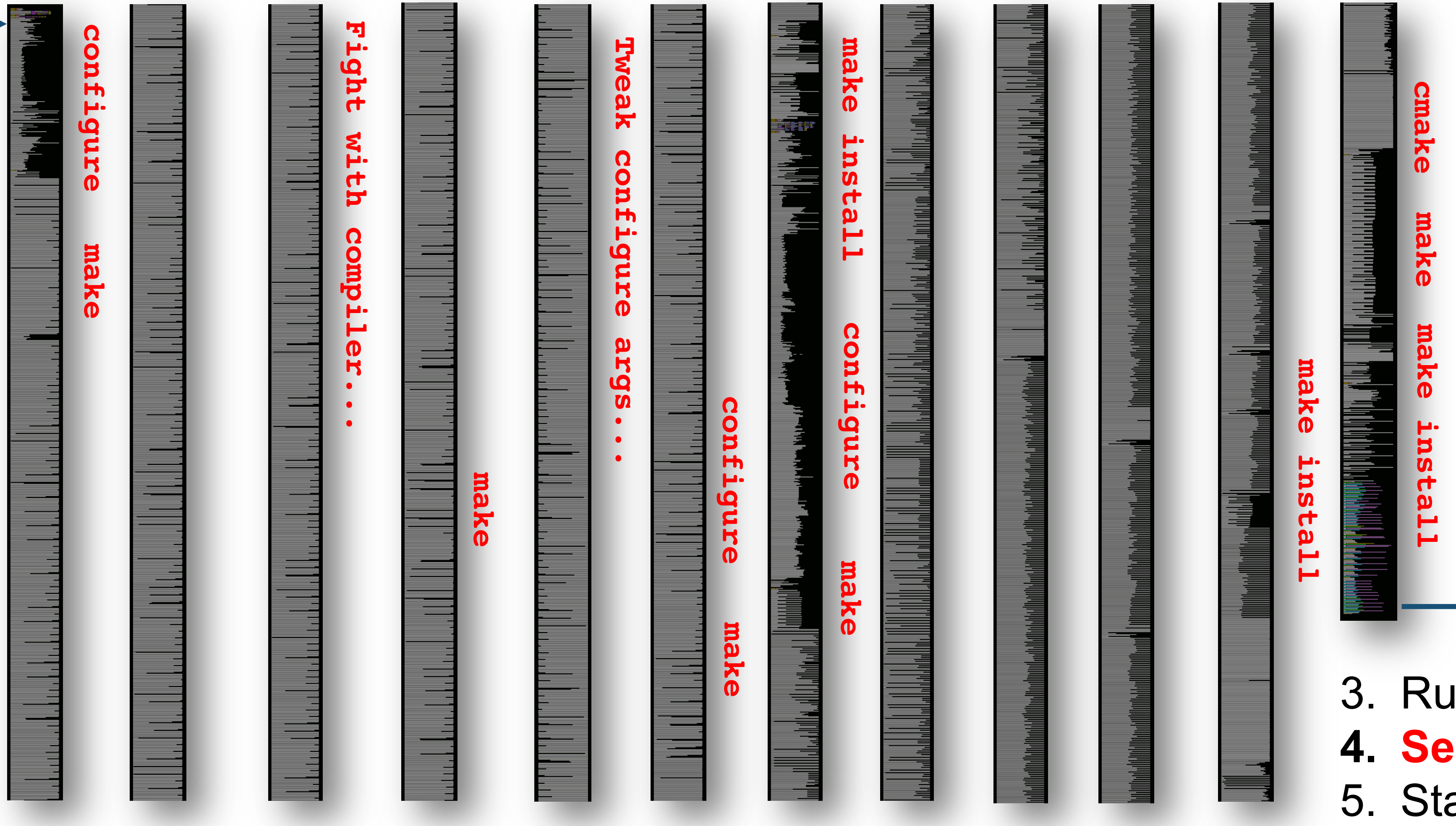
= up to **1,260,000** combinations!

- Every application has its own stack of dependencies.
- Developers, users, and facilities dedicate (many) FTEs to building & porting.
- Often trade reuse and usability for performance.

We must make it easier to rely on others' software!

How to install software on a supercomputer

1. Download all 16 tarballs you need
2. Start building!



3. Run code
4. **Segfault!?**
5. Start over…

What about modules?

- Most supercomputers deploy some form of *environment modules*
  - TCL modules (dates back to 1995) and Lmod (from TACC) are the most popular

```
$ gcc
- bash: gcc: command not found

$ module load gcc/7.0.1
$ gcc -dumpversion
7.0.1
```

- Modules don’t handle installation!
  - They only modify your environment (things like PATH, LD_LIBRARY_PATH, etc.)
- Someone (likely a team of people) has already installed gcc for you!
  - Also, you can *only* `module load` the things they’ve installed

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

@spackpm

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                                # set up compilers
$ spack external find                                # set up external packages
```

```
$ spack install tau                                                      unconstrained
$ spack install tau@2.30.1                                               @ custom version
$ spack install tau@2.30.1 %gcc@7.3.0                                    % custom compiler
$ spack install tau@2.30.1 %gcc@7.3.0 +level_zero                        +/- build option
$ spack install tau@2.30.1 %gcc@7.3.0 +mpi ^mvapich2@2.3~wrapperrpath ^ 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

EXASCALE COMPUTING PROJECT

`spack find` shows what is installed

```
Singularity> spack find
==> 319 installed packages
-- linux-ubuntu18.04-power9le / gcc@7.3.0 -----------------------
autoconf@2.69     diffutils@3.7     libiconv@1.16      m4@1.4.18     ncurses@6.2             openssl@1.1.1g      texinfo@6.5
automake@1.16.2   findutils@4.6.0   libpciaccess@0.16  matio@1.5.17  netcdf-c@4.7.4          parmetis@4.0.3      trilinos@13.0.0
boost@1.74.0      glm@0.9.7.1       libsigsegv@2.12    metis@5.1.0   netlib-scalapack@2.1.0  perl@5.26.1         util-macros@1.19.1
bzip2@1.0.8       hdf5@1.10.7       libtool@2.4.6      mpich@3.2.1   omega-h@9.29.0          pkgconf@1.7.3       xz@5.2.5
cmake@3.18.4      hypre@2.20.0      libxml2@2.9.10     mumps@5.3.3   openblas@0.3.10         suite-sparse@5.7.2  zlib@1.2.11

-- linux-ubuntu18.04-ppc64le / gcc@7.3.0 -------------------------
adiak@0.1.1              flit@2.1.0               libpfm4@4.11.0              papyrus@develop               py-more-itertools@7.2.0     qthreads@1.14
adios@1.13.1             gasnet@2020.3.0          libpng@1.6.37               parallel-netcdf@1.12.1        py-mpi4py@3.0.3             raja@0.12.1
adios2@2.6.0             gasnet@2020.3.0          libpthread-stubs@0.4        parmetis@4.0.3                py-nbclient@0.5.0           rankstr@0.0.2
adlbx@0.9.2              gdbm@1.18.1              libquo@1.3.1                pcre@8.44                     py-nbconvert@6.0.1          readline@8.0
aml@0.1.0                gettext@0.20.2           libsigsegv@2.12             pcre2@10.35                   py-nbformat@5.0.7           redset@0.0.3
amrex@20.10              gettext@0.21             libsodium@1.0.18            pdsh@2.31                     py-nest-asyncio@1.4.0       rempi@1.1.0
arborx@0.9-beta          ginkgo@1.3.0             libtool@2.4.6               pdt@3.25.1                    py-notebook@6.1.4           scr@2.0.0
argobots@1.0             git@2.28.0               libunistring@0.9.10         perl@5.26.1                   py-numpy@1.19.2             shuffile@0.0.3
arpack-ng@3.7.0          git@2.28.0               libunwind@1.4.0             petsc@3.13.6                  py-oauthlib@3.1.0           slate@develop
ascent@develop           glm@0.9.7.1              libunwind@1.4.0             petsc@3.14.0                  py-pamela@1.0.0             slepc@3.14.0
autoconf@2.69            globalarrays@5.7         libuuid@1.0.3               pkgconf@1.7.3                 py-pandocfilters@1.4.2      snappy@1.1.8
automake@1.16.2          gmake@4.2.1              libxml2@2.9.10              plasma@20.9.20                py-parso@0.6.1              sqlite@3.31.1
axl@0.3.0                gmp@6.1.2                libyogrt@1.24               precice@2.1.1                 py-petsc4py@3.13.0          strumpack@5.0.0
axom@0.3.3               googletest@1.10.0        libzmq@4.3.2                pumi@2.2.2                    py-pexpect@4.7.0            suite-sparse@5.7.2
bash@5.0                 gotcha@0.0.2             lmod@8.3                    py-alembic@1.0.7              py-pickleshare@0.7.5        sundials@5.4.0
binutils@2.33.1          gotcha@1.0.3             lua@5.3.5                   py-argon2-cffi@20.1.0         py-prometheus-client@0.7.1  superlu@5.2.1
bmi@develop              gperftools@2.7           lua-luafilesystem@1_7_0_2   py-asn1crypto@0.24.0          py-prompt-toolkit@2.0.9     superlu-dist@6.3.0
bolt@1.0                 hdf5@1.8.21              lua-luaposix@33.4.0         py-async-generator@1.10       py-psutil@5.7.2             superlu-dist@6.3.1
boost@1.73.0             hdf5@1.8.21              lwgrp@1.0.3                 py-attrs@19.3.0               py-ptyprocess@0.6.0         swig@4.0.2
boost@1.73.0             hdf5@1.10.6              lz4@1.9.2                   py-babel@2.7.0                py-py@1.8.0                 sz@1.4.12.3
boost@1.73.0             hdf5@1.10.6              lzo@2.10                    py-backcall@0.1.0             py-pycparser@2.20           sz@2.0.2.0
boost@1.73.0             hpctoolkit@2020.08.03    m4@1.4.18                   py-bleach@3.1.0               py-pyelftools@0.26          sz@2.1.10
butterflypack@1.2.0      hpx@1.5.1                magma@2.5.4                 py-blinker@1.4                py-pygments@2.6.1           tar@1.32
bzip2@1.0.8              hwloc@1.11.11            margo@0.4.3                 py-certifi@2020.6.20          py-pyjwt@1.7.1              tasmanian@7.3
c-blosc@1.17.0           hwloc@2.2.0              matio@1.5.17                py-certipy@0.1.3              py-pyopenssl@19.0.0         tau@2.29
caliper@2.4.0            hypre@2.18.2             mbedtls@2.16.7              py-cffi@1.14.3                py-pyrsistent@0.15.7        tcl@8.6.10
cinch@master             hypre@2.20.0             mercury@1.0.1               py-chardet@3.0.4              py-pytest-runner@5.1        texinfo@6.5
cmake@3.17.3             intel-tbb@2020.3         mercury@1.0.1               py-cryptography@2.7           py-python-dateutil@2.8.0    turbine@1.2.3
conduit@master           kokkos@3.2.00            metis@5.1.0                 py-cython@0.29.21             py-python-editor@1.0.4      umap@2.1.0
conduit@master           kokkos-kernels@3.2.00    mfem@4.1.0                  py-decorator@4.4.2            py-python-oauth2@1.1.1      umpire@4.0.1
cuda@10.2.89             kvtree@1.0.2             mpark-variant@1.4.0         py-defusedxml@0.6.0           py-pytz@2020.1              umpire@4.0.1
curl@7.72.0              legion@20.03.0           mpich@3.2.1                 py-entrypoints@0.3            py-pyzmq@18.1.0             unifyfs@0.9.0
darshan-runtime@3.2.1    leveldb@1.22             mpifileutils@develop        py-idna@2.8                   py-requests@2.24.0          unzip@6.0
da[illegible]3.2.1       libarchive@3.4.1         mumps@5.3.3                 py-importlib-metadata@2.0.0   py-send2trash@1.5.0         upcxx@2020.3.0
di[illegible]s@3.7       libbsd@0.10.0            ncurses@6.2                 py-ipykernel@5.3.4            py-setuptools@50.1.0        util-macros@1.19.1
```

All the versions coexist!

- Multiple versions of same package are ok.

Packages are installed to automatically find correct dependencies.

Binaries work *regardless of user's environment*.

Spack also generates module files.

- Don't *have* to use them.

The Spack community is growing rapidly

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

Visit spack.io

github.com/spack/spack

@spackpm

ECP SW Technology Software Architecture – SDKs

ECP ST SDKs will span all technology areas

Motivation: Properly chosen cross-team interactions will build relationships that support interoperability, usability, sustainability, quality, and productivity within ECP ST.

Action Plan: Identify product groupings where coordination across development teams will improve usability and practices, and foster community growth among teams that develop similar and complementary capabilities.

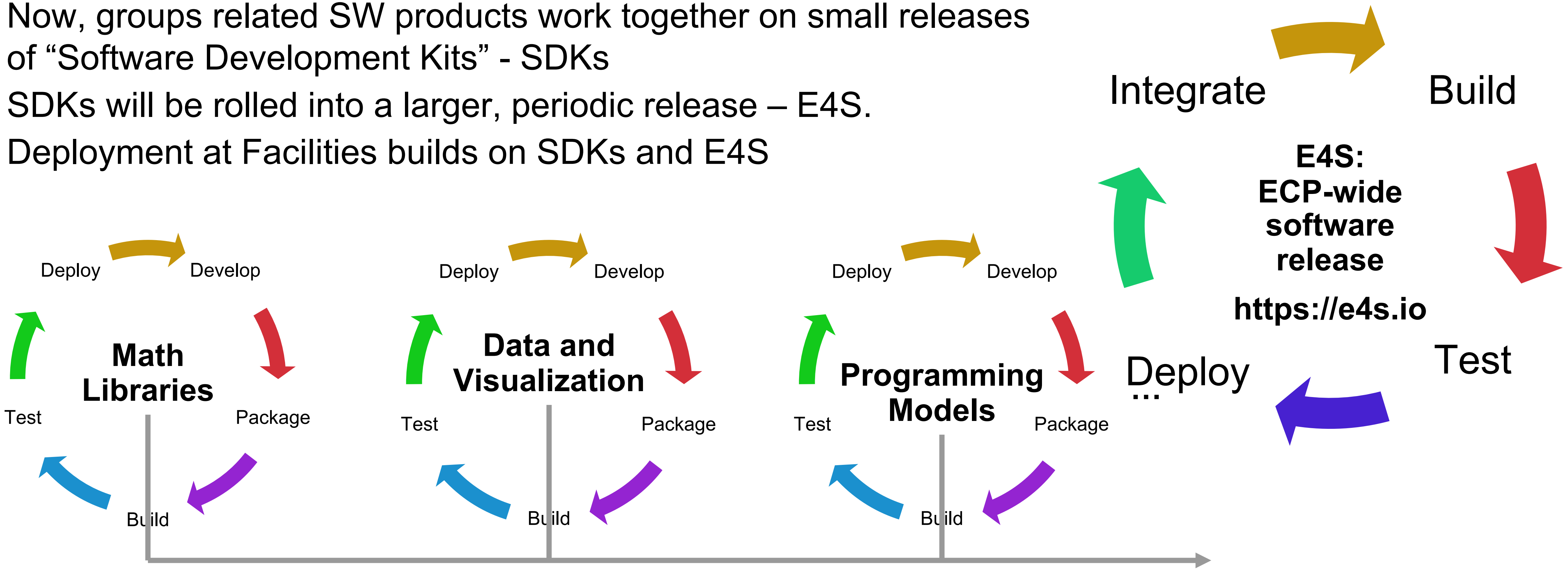
PMR Core (17)	Compilers and Support (7)	Tools and Technology (11)	xSDK (16)	Visualization Analysis and Reduction (9)	Data mgmt, I/O Services, Checkpoint restart (12)	Ecosystem/E4S at-large (12)
QUO	openarc	TAU	hypre	ParaView	SCR	mpiFileUtils
Papyrus	Kitsune	HPCToolkit	FleSCI	Catalyst	FAODEL	TriBITS
SICM	LLVM	Dyninst Binary Tools	MFEM	VTK-m	ROMIO	MarFS
Legion	CHiLL autotuning comp	Gotcha	Kokkoskernels	SZ	Mercury (Mochi suite)	GUFI
Kokkos (support)	LLVM openMP comp	Caliper	Trilinos	zfp	HDF5	Intel GEOPM
RAJA	OpenMP V & V	PAPI	SUNDIALS	VisIt	Parallel netCDF	BEE
CHAI	Flang/LLVM Fortran comp	Program Database Toolkit	PETSc/TAO	ASCENT	ADIOS	FSEFI
PaRSEC*		Search (random forests)	libEnsemble	Cinema	Darshan	Kitten Lightweight Kernel
DARMA		Siboka	STRUMPACK	ROVER	UnifyCR	COOLR
GASNet-EX		C2C	SuperLU		VeloC	NRM
Qthreads		Sonar	ForTrilinos		IOSS	ArgoContainers
BOLT			SLATE		HXHIM	Spack
UPC++			MAGMA			
MPICH			DTK			
Open MPI			Tasmanian			
Umpire			TuckerMPI			
AML						

Legend
- PMR
- Tools
- Math Libraries
- Data and Vis
- Ecosystems and delivery

EXASCALE COMPUTING PROJECT

ECP is working towards a periodic, hierarchical release process

- In ECP, teams increasingly need to ensure that their libraries and components work together
  - Historically, HPC codes used very few dependencies
- Now, groups related SW products work together on small releases of “Software Development Kits” - SDKs
- SDKs will be rolled into a larger, periodic release – E4S.
- Deployment at Facilities builds on SDKs and E4S



E4S Components

- E4S is a curated release of ECP ST products based on Spack [http://spack.io].
- E4S Spack cache to support bare-metal installs at facilities and custom container builds:
  - x86_64, ppc64le, and aarch64
- Container images on DockerHub and E4S website of pre-built binaries of ECP ST products.
- Base images and full featured containers with support for GPUs.
- GitHub recipes for creating custom images from base images.
- e4s-cl for container launch and for replacing MPI in application with system MPI libraries.
- Validation test suite on GitHub provides automated build and run tests.
- Automates build process via GitLab Continuous Integration to ensure packages can be built.
- E4S Doc Portal aggregates and summarizes documentation and metadata by raking product repos.
- E4S VirtualBox image with support for Docker, Shifter, Singularity, and Charliecloud runtimes.
- AWS image to deploy E4S on EC2.
- GCP image to deploy E4S on GCP.

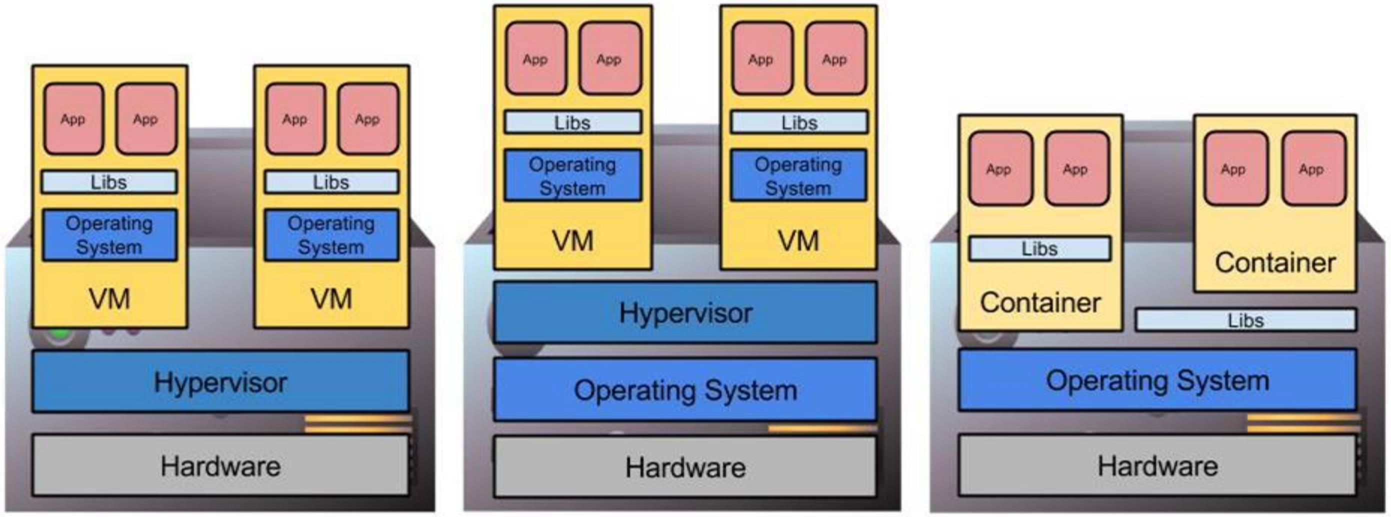
https://e4s.io

What are containers

- A lightweight collection of executable software that encapsulates everything needed to run a single specific task
  - Minus the OS kernel
  - Based on Linux only
- Processes and all user-level software is isolated
- Creates a portable* software ecosystem
- Think `chroot` on steroids
- Docker most common tool today
  - Available on all major platforms
  - Widely used in industry
  - Integrated container registry via Dockerhub

Hypervisors and Containers

Type 1 hypervisors insert layer below host OS
Type 2 hypervisors work as or within the host OS
Containers do not abstract hardware, instead provide “enhanced chroot” to create isolated environment
Location of abstraction can have impact on performance
All enable custom software stacks on existing hardware



Type 1 Hypervisor **Type 2 Hypervisor** **Containers**

Download E4S 2021-02 GPU Container Image

https://e4s-project.github.io/download.html

`# docker pull ecpe4s/ubuntu18.04-e4s-gpu`

RHEL 7

SPACK MINIMAL

ecpe4s/rhel7-spack docker GitHub

E4S COMPREHENSIVE

ecpe4s/rhel7-e4s docker GitHub

CUSTOM

ecpe4s/superlu_sc docker GitHub

Ubuntu 18.04

E4S GPU IMAGE

- ecpe4s/ubuntu18.04-e4s-gpu docker GitHub

x86_64 version: CUDA and ROCM

ppc64le version: CUDA

SPACK MINIMAL

ecpe4s/ubuntu18.04-spack docker GitHub

E4S COMPREHENSIVE

ecpe4s/ubuntu18.04-e4s docker GitHub

CentOS 7

SPACK MINIMAL

ecpe4s/centos7-spack docker GitHub

E4S COMPREHENSIVE

ecpe4s/centos7-e4s docker GitHub

CUSTOM

- - - -

https://e4s.io

E4S v2021-02 GPU Release for x86_64

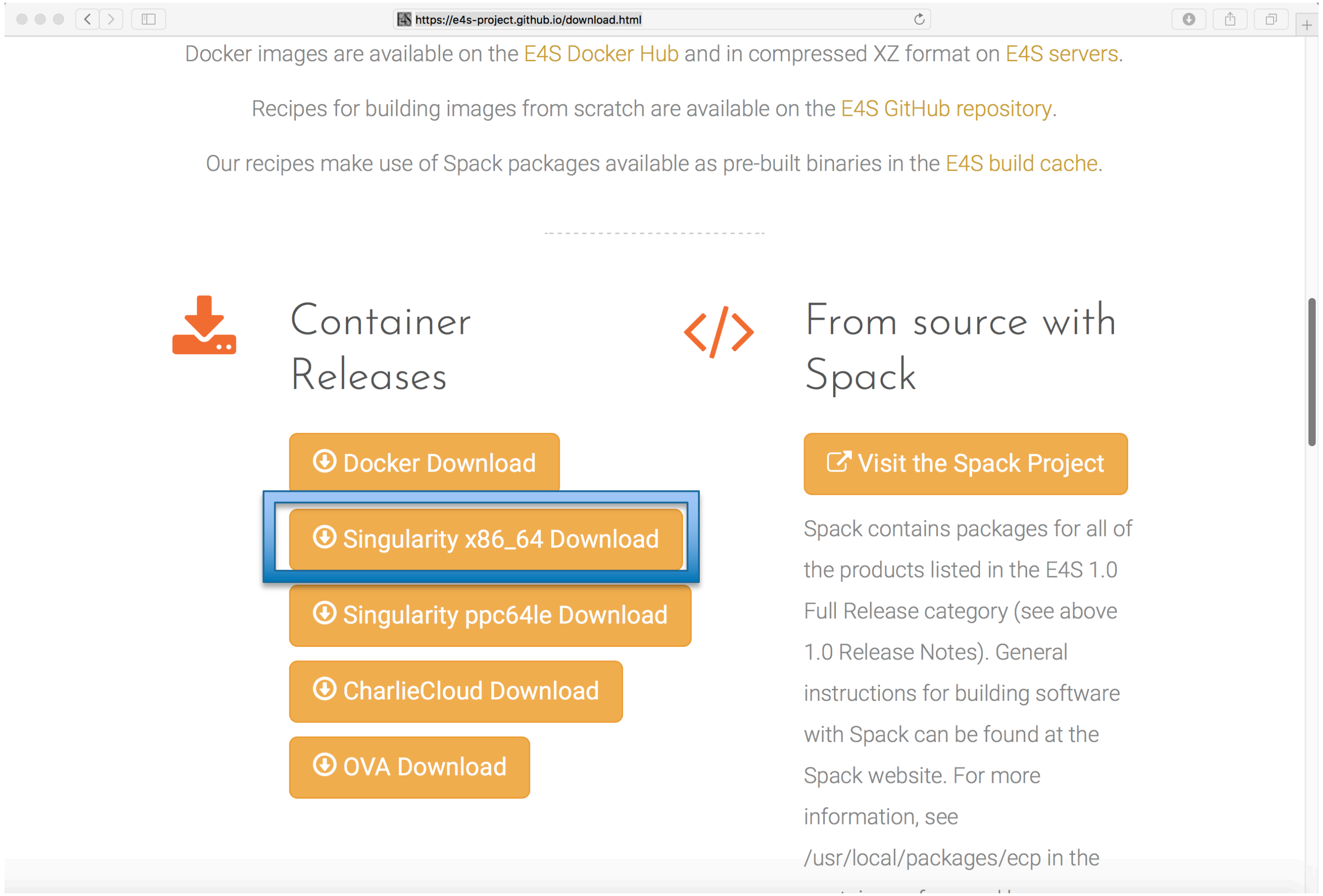
```
1: adios2       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/adios2-2.6.0-nkp24j7enorn3dt7626chuqm3pbkrvfe
2: aml  /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/aml-0.1.0-3mwyb6cf6ervfnruqb5u33v46buyuqth
3: arborx       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/arborx-0.9-beta-qjzxlkcgplto6pnpjwejoh5xpoik3adr
4: argobots     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/argobots-1.0-yoafg2slps7kp4dkmb6pzu5z2a37sgs4
5: ascent       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/ascent-develop-ciwqg6lh6unww3hjsnu47wr7cpqptqgy
6: axom         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/axom-0.3.3-tzyejxpy3p3ekaev35k2bhpkr74cnuhh
7: bolt         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/bolt-1.0-uxku5w5qdfnpa4atgzcbraq7wop7lunc
8: caliper      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/caliper-2.4.0-lfdx3gc6qodg2abbpovib3thdsmsamnn
9: darshan-runtime      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/darshan-runtime-3.2.1-jqugqxx2uunyaduoe3owhd2snves6mlr
10: dyninst     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/dyninst-10.2.1-xad3v6rvosm6qfai5fc7d4nn33svtzzf
11: faodel      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/faodel-1.1906.1-ijilel2vjionmj56mscqkw2hpecfsuym
12: flecsi      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/flecsi-1-c7sevlnc2ak4pf2jgq6wh3mwictch5l2
13: flit        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/flit-2.1.0-yvvog7kmax22ei2yyrwfxj3heinmz5am
14: gasnet      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/gasnet-2020.3.0-ufrq5hym67eq3jsg4jtttjjqgo4i6hnq
15: ginkgo      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/ginkgo-1.2.0-r6lorgchpr5qrcwyqqxtewqdhtpi4rmt
16: globalarrays        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/globalarrays-5.7-bow6d32j63j6gusotzjuityznwqvv64b
17: gotcha      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/gotcha-1.0.3-7n7bjnzsnf5w5tnihiok3otbaewdhjmu
18: hdf5        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/hdf5-1.10.6-k74avubedd5knvlc73dr3ib5oyw6bcwn
19: hpctoolkit  /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/hpctoolkit-2020.08.03-wck4g3h3jhfvzvxorelxqunbe3xsesry
20: hpx         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/hpx-1.5.0-pynmocntkmuwkowyo5jxtycvg34w6kue
21: hypre       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/hypre-2.19.0-vqo72wn6ei7ruitpg7drkje2rdbdfguo
22: kokkos      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/kokkos-3.2.00-pqv3uugd6cv3qftyur3rx6dm2gao2tg3
23: kokkos-kernels      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/kokkos-kernels-3.1.00-y4veufypftworlbehxusg4yzh6n7anhp
24: legion      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/legion-20.03.0-zkbz7h2wuze4dgbwcbo4w5fvqltugmog
25: libnrm      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/libnrm-0.1.0-kp5jb7o4kow25rnggiditwtmdbeebojs
26: libquo      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/libquo-1.3.1-w45wcw6dqbiajeeauj3ryaeskqu7bzx6
27: magma       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/magma-2.5.3-yksxthffslhjrhzwgcx7smz2tca6ojfn
28: mercury     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/mercury-1.0.1-ppledsr3drk2upciytfsuawfxrtjp73q
29: mfem        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/mfem-4.1.0-kivaike2qintplgufwp5yf2mj3n36ay3
30: mpich       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/mpich-3.2.1-kgwtpelzobpkrvg24ct6padfbhw7nene
31: mpifileutils        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/mpifileutils-develop-djje5g7ts55g3yic3bms426c2zi7gqsj
32: ninja       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/ninja-1.10.1-7zbbtuslw25nmqo4ur6abyyf3tchnqvv
33: omega-h     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/omega-h-9.29.0-eln73w7ytpvgqtkmkqyjm4gsabsu2w4p
```

- 67 ECP ST products
- Ubuntu v18.04 x86_64
- AI/ML package support
  - TensorFlow 2.3.5
  - PyTorch 1.8
  - Horovod
- Support for GPUs
  - AMD ROCm 3.8
  - NVIDIA CUDA 10.2, 11
  - Intel OneAPI 2021.1
- Kokkos with support for AMD GPUs!

E4S v2021-02 GPU Release for x86_64

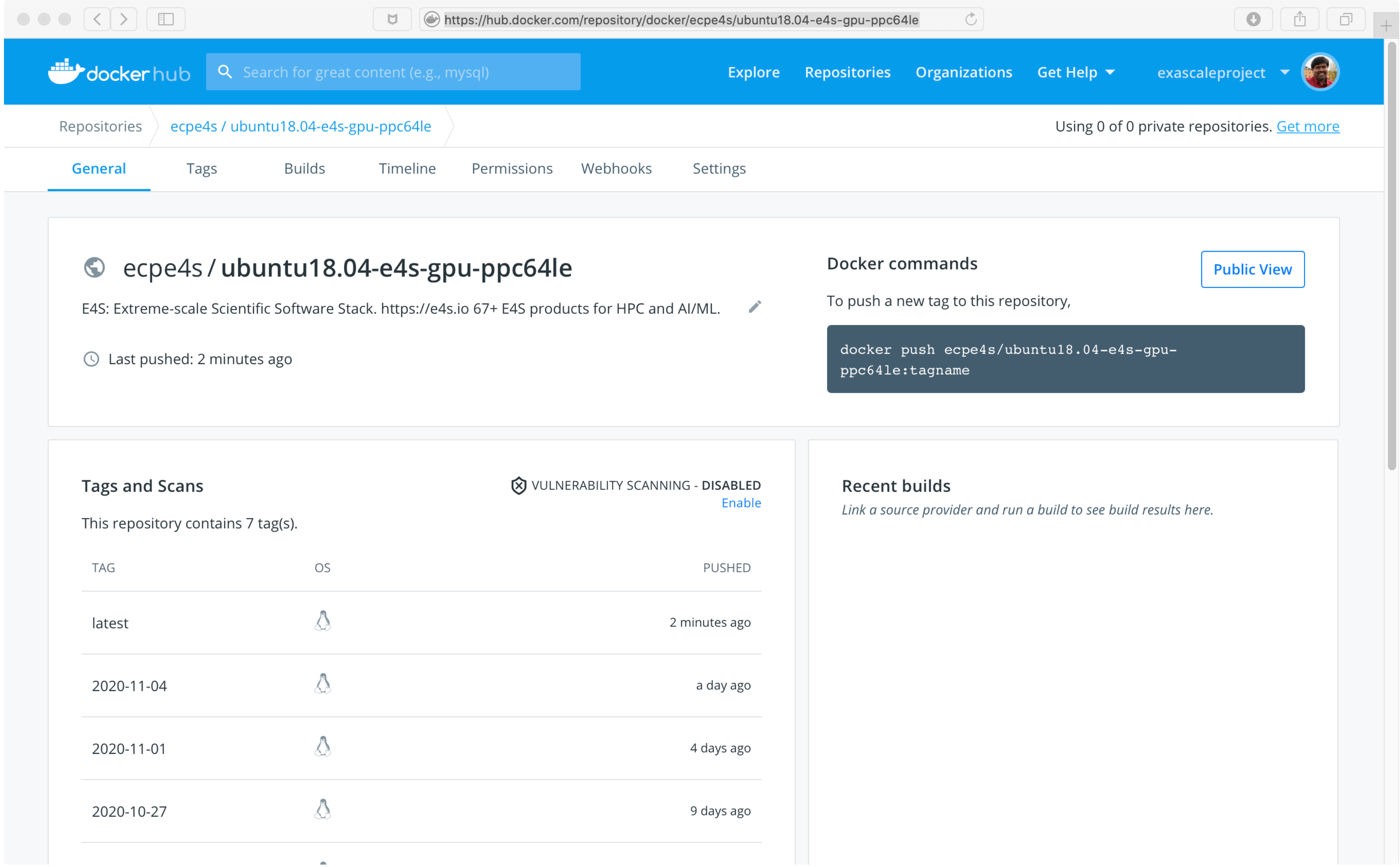
```
34: openpmd-api         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/openpmd-api-0.12.0-4myph6pbjnupgupxdlvbxvqqeqx6atyp
35: openmpi     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/openmpi-3.1.6-6yqtoym56as6xso2pdgkmn4bcsoyufku
36: papi        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/papi-6.0.0.1-gorrfrvrik575lldzgq46qmmu63kxl7x
37: papyrus     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/papyrus-develop-iu3dgpmwwyykgv5mpw2dwcrol4wbwbai
38: parallel-netcdf     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/parallel-netcdf-1.12.1-tmmkzibn43xr7su76msxxusyzrphdtn5
39: pdt         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/pdt-3.25.1-kvi5wuu5y72fypijti3nxqvdn7zpj6ni
40: petsc       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/petsc-3.13.4-llg3u4rrt5axrqlim75tt73epewxu4fb
41: plasma      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/plasma-19.8.1-tji7bojb5ne5hqj2mwn5bqq2tfkm23ke
42: precice     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/precice-2.1.0-ozdmbat2hlivccha3nklbeahikgynewu
43: pumi        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/pumi-2.2.2-52czzdbxeg7pmjkd55nub5jgxzodcprh
44: py-jupyterhub       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/py-jupyterhub-1.0.0-tr3wcolaij3kbzb6xm4mbbvakcstwsw3
45: py-libensemble      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/py-libensemble-0.7.0-mxvqxhiiblqnmhlfepbxboyiskqyvbej
46: qthreads    /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/qthreads-1.14-neshsclplh7ttkebm34grztaijqohnxt
47: raja        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/raja-0.11.0-w25bj2dys6cjqn7isgcjfyvte3tuulev
48: rempi       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/rempi-1.1.0-sideqdbiik2yseshs3loh4sictbis3t6
49: scr         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/scr-2.0.0-yh3chyq5gayuk6r4juejjiye6zg3rh3u
50: slate       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/slate-develop-jnysy2rh5vxhwua5ubtvq4bsfd3py7d5
51: slepc       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/slepc-3.13.4-q3lalpbqoshiyvjjgrnhb2iqiisvnfrp
52: strumpack   /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/strumpack-4.0.0-rlbti5eqc5rjhfisxv2uxevj6m3fn5gg
53: sundials    /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/sundials-5.3.0-3g52gh4a6h4ohucqart5i4m6pi66woj6
54: superlu-dist        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/superlu-dist-6.3.1-o2hkund66coxn2rrbtlalda2vq35uu7j
55: stc         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/stc-0.8.3-oxfik7nsmgufoqyy7xilzsrct7it63ej
56: swig        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/swig-4.0.1-htxmzjd5sed5yfibw6j7jn5cx6p7g72x
57: sz  /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/sz-2.1.9-tcatyiuzh6quctrgd2g3dcli7xa7gvtj
58: tasmanian   /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/tasmanian-7.1-quo3grs5kb2xrvjufpi7vn66cpjfnadv
59: tau         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/tau-2.29-ijw2nbphmlfkt42ubwz7g5a5yru22ikn
60: trilinos    /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/trilinos-13.0.0-6xfnp44g5xm7gpn2en6gkwzfceykfd3x
61: turbine     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/turbine-1.2.3-q4qjvgxjl3cbuyquo6zrurb4mwfn6wkp
62: umap        /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/umap-2.0.0-5tob3exzrmwoitudu5pstbb2dms3xnto
63: umpire      /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/umpire-3.0.0-6woo2uuvazcucxikc6xad6g3zksu2ygi
64: unifyfs     /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/unifyfs-0.9.0-be7mqbng7kdeewdlgvlhdm4jkxnquiil
65: upcxx       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/upcxx-2020.3.0-pshe62qyvmnrvesqa4pkj6bdq3fxxucf
66: veloc       /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/veloc-1.4-gk3iwfjhmgkwlawp7rmxf2eh37rqpqm2
67: zfp         /opt/spack/opt/spack/linux-ubuntu18.04-x86_64/gcc-7.5.0/zfp-0.5.5-6r6yaco7gga5w4gbuvid3zt2iohrnepj
```

E4S Support for Singularity Container Runtime [Sylabs.io]



- wget http://tau.uoregon.edu/ecp.simg; singularity run ./ecp.simg
- singularity run ecp.simg
- Supports Intel OneAPI, CUDA, and ROCm
- spack find

E4S v2021-02 Release: GPU, ppc64le for Docker Containers



- 67 ECP Products
- Support for GPUs
  - NVIDIA (CUDA 10.2)
  - ppc64le and x86_64

% docker pull ecpe4s/ubuntu18.04-e4s-gpu

EXASCALE COMPUTING PROJECT

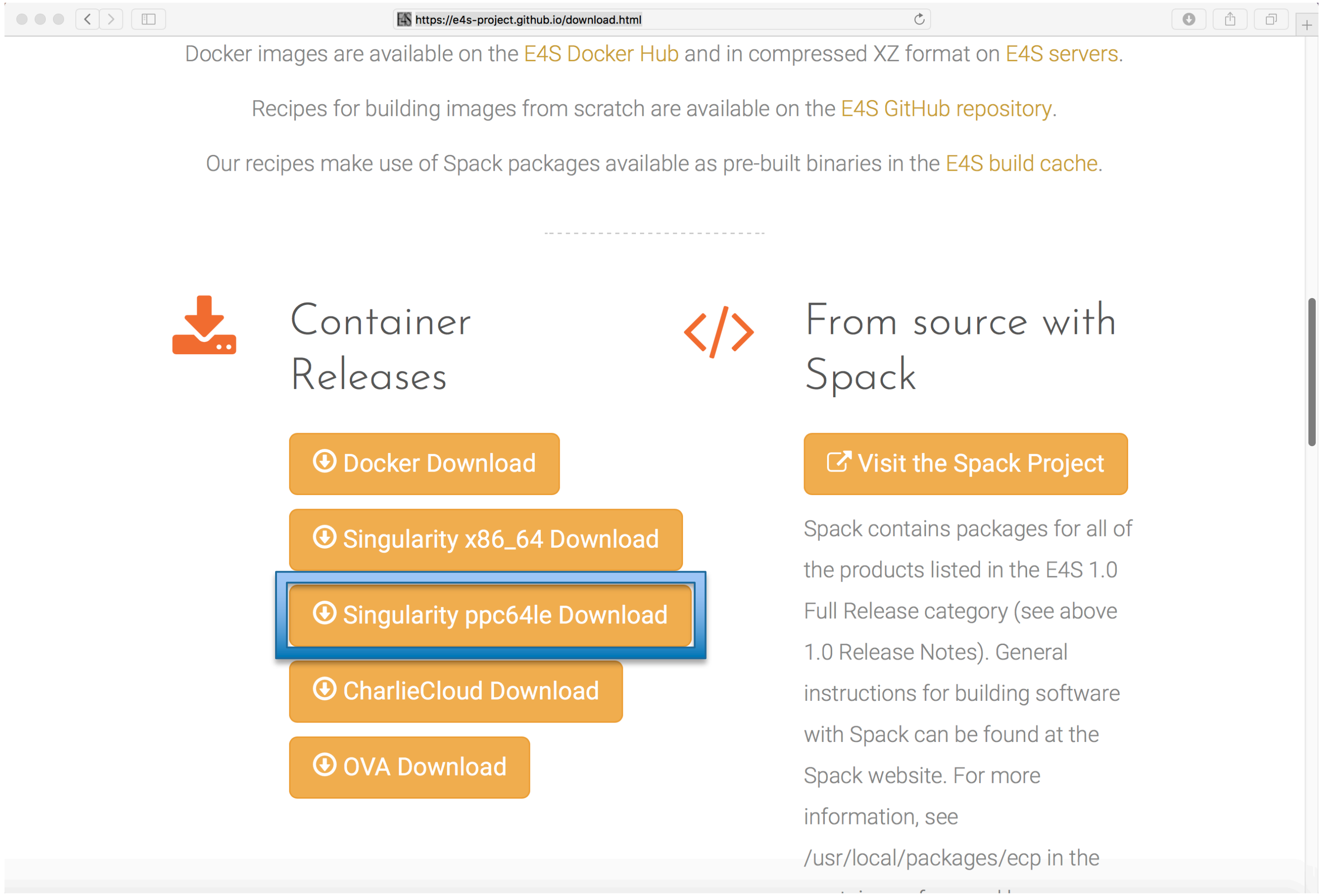
E4S v2021-02 GPU Release: 67 E4S Products (ppc64le)

```
1: adios2       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/adios2-2.6.0-veoqi5iqkx4kbeddhxoroggvxqqbtvos
2: aml  /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/aml-0.1.0-ftizegmvpbweuyzg75g3ndzhdyjx37op
3: amrex        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/amrex-20.10-4z5quvlqt3fbzv5n6rrjv5byq7472emy
4: arborx       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/arborx-0.9-beta-p7lw7eobsrdpqwhb7ispxgphng2tn4nt
5: ascent       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/ascent-develop-7ktzsmvluqvd4xzoop7hjwddyjetn2ai
6: axom         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/axom-0.3.3-zfgqs6qa6vxlodjnaojeffmyl26czmp5
7: argobots     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/argobots-1.0-qra2gqxuisqqlbdfrhwm5mvq2iga3l3l
8: bolt         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/bolt-1.0-ojy67rk47pcbqpcvuq6a4c7g7qysvndv
9: caliper      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/caliper-2.4.0-6xzehuxs2updvdrl2tdvcym3n6nf3y3l
10: darshan-runtime     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/darshan-runtime-3.2.1-6uzihv7v75yu47c2jca4qpxqdtgptn2g
11: dyninst     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/dyninst-10.2.1-jvqx4j3ehuh73pp67b4vdy4co3kivma5
12: faodel      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/faodel-1.1906.1-r77asm5xkb256omn4trg5hnxc3e376uy
13: flecsi      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/flecsi-1-2kxukdrijujvbmsabmmj3um54ukhrayk
14: flit        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/flit-2.1.0-tepzltg6kmefdg4eo2rbzwmjeca56bmc
15: gasnet      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/gasnet-2020.3.0-uynuhs6itzczkfpgbnlm2xgotvqmmeb6
16: ginkgo      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/ginkgo-1.3.0-dodvdbixjpdg5ci5xrgomjeqybiob33i
17: globalarrays        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/globalarrays-5.7-3zbsvrakwto5jc454jl3l36rpvray25h
18: gotcha      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/gotcha-1.0.3-pvjdzcg3fggpagjcsorwidsllflmomnz
19: hdf5        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/hdf5-1.10.6-arkkhmy4auglzqndt7xraupyvgkrpv7o
20: hpctoolkit  /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/hpctoolkit-2020.08.03-yqayfprp2aleaxtzq543c75lcvcviso7
21: hpx         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/hpx-1.5.1-tzfs3nkglsacequjxflokigwgjzabybk
22: hypre       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/hypre-2.20.0-ewmv445dkzmju4upg4rregq7apgkcdbu
23: kokkos      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/kokkos-3.2.00-3qzjrzoxl5lpqgtaq4atid6ylgkko3uk
24: kokkos-kernels      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/kokkos-kernels-3.2.00-n4trpqubmexqahdy4tolj6nhfml5j4v6
25: legion      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/legion-20.03.0-xsotehq7eg77hcguvqx5qymfhimgtuic
26: libnrm      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/libnrm-0.1.0-q67khfosljacbl3djdg5jeh4thsl5p5f
27: libquo      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/libquo-1.3.1-syjf6c3adia34wlwneacynrwkhh72i3u
28: magma       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/magma-2.5.4-fzeektdrkybbuo6i6niikzglcwlnt2jx
29: mercury     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/mercury-1.0.1-ufxkkvb7osjnwgbfevdhtrmtuoj6dfbz
30: mfem        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/mfem-4.1.0-qrepufdzopbphsyuyc6npn7k2tpprd5w
31: mpich       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/mpich-3.2.1-5m7ofmtvtov45hcudrm3qvd2dyheunyv
32: mpifileutils        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/mpifileutils-develop-rd5xj2pmx5vdd7fddrhbrvn2uykg4uay
33: ninja       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/ninja-1.10.1-cr2ada5fjgvkvdtmxel4zj6venfiif5e
```

E4S v2021-02 GPU Release: 67 E4S Products (ppc64le)

```
34: omega-h     /opt/spack/opt/spack/linux-ubuntu18.04-power9le/gcc-7.3.0/omega-h-9.29.0-ziz55mnp5r7l4kuhx4zqmjp2imjdvrk5
35: openmpi     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/openmpi-3.1.6-utceq6uech6rgnabxevau4lhtzrwbaol
36: openpmd-api         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/openpmd-api-0.12.0-szt65gmfb76iwdbcfkhryfztg5jwjd7g
37: papi        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/papi-6.0.0.1-xu35qtffffq2ofyjic3fafmj6yeijoih
38: papyrus     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/papyrus-develop-2zopf6p3ha4v7ijxslxskrf2qyhpt3py
39: parallel-netcdf     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/parallel-netcdf-1.12.1-svuejkorgi2bzvhgq4wts72bcjfn426r
40: pdt         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/pdt-3.25.1-opxwliyf5vqgt3hbla7qspf3laaqbt74
41: petsc       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/petsc-3.14.0-phaqc52ryvhcib37qqjg2lmqdebql2uo
42: plasma      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/plasma-20.9.20-vc4olrzgwsvqx7mevom2j7mhsgb6ynam
43: precice     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/precice-2.1.1-qlitin5qdhtz3n7rg4jjzxkdss4qocvn
44: pumi        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/pumi-2.2.2-m4uipa7yh632dftix4kzyxcz3pm3fasv
45: py-jupyterhub       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/py-jupyterhub-1.0.0-gzzlya6f4gr2xgsgpndmbp2pkffm3tuc
46: py-libensemble      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/py-libensemble-0.7.1-oee4zlxigkjc5nnkr6fyu7thzsnftvvu
47: py-petsc4py         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/py-petsc4py-3.13.0-g2rp2v37qbpx5fo5fmg6c4xtrj6shsbz
48: qthreads    /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/qthreads-1.14-bdxplr2gf7knpek4vo5sjvzh5py5fdaf
49: raja        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/raja-0.12.1-q32nuxmeowavkwzmoiwx6f5md246tw66
50: rempi       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/rempi-1.1.0-h3x5q2rwwsv34v7e4ricjw65wcd5mvkg
51: scr         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/scr-2.0.0-2okrlxki5b63gzakjy2x4sbovrmegmcx
52: slate       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/slate-develop-2jp7v35nifhyucbf4vmi3mjsernm5t26
53: slepc       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/slepc-3.14.0-7qn6k5qxzf32tc2cnuk2mknlvqvv6hfw
54: strumpack   /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/strumpack-5.0.0-gy5opc36suubh6uoiqy4l223psdyrilg
55: sundials    /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/sundials-5.4.0-wonraynurs6xhyv6m6bc7o4grlwchlnp
56: superlu-dist        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/superlu-dist-6.3.1-poufv43kq7tw2rw6upldbpcpabkpbdta
57: swig        /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/swig-4.0.2-3bddrfojvkrowa43v5so3ongbmhzxx5s
58: sz  /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/sz-2.1.10-fld5xazn2spjg46yaaaam5gftgyb5loa
59: tasmanian   /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/tasmanian-7.3-zbz26kn2yabritfi2wsbqv5raexgi4p3
60: tau         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/tau-2.29-zqbkmoraislptbdny6fw4pakoipm3cbv
61: trilinos    /opt/spack/opt/spack/linux-ubuntu18.04-power9le/gcc-7.3.0/trilinos-13.0.0-olf4mdmym4sjbgue66gx42k7dbeb6z27
62: turbine     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/turbine-1.2.3-jy42tjmn7rd2ofwwb3jaanlri2hnte65
63: umpire      /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/umpire-4.0.1-ynagdhefpcujnpeybxtasoqecr2p7bxj
64: unifyfs     /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/unifyfs-0.9.0-sxswh3b5upcys4bxc5wdzczvwxvn6emg
65: upcxx       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/upcxx-2020.3.0-i6hf7mat23um3fz5wexqswvn6mm4o7zp
66: veloc       /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/veloc-1.4-7ygadmpwv2zr26ec6opicysts4mxkwym
67: zfp         /opt/spack/opt/spack/linux-ubuntu18.04-ppc64le/gcc-7.3.0/zfp-0.5.5-3r4a4s3qdeqbdabvlwlswrgig62yc6yj
```

E4S Support for Singularity Container Runtime [Sylabs.io]



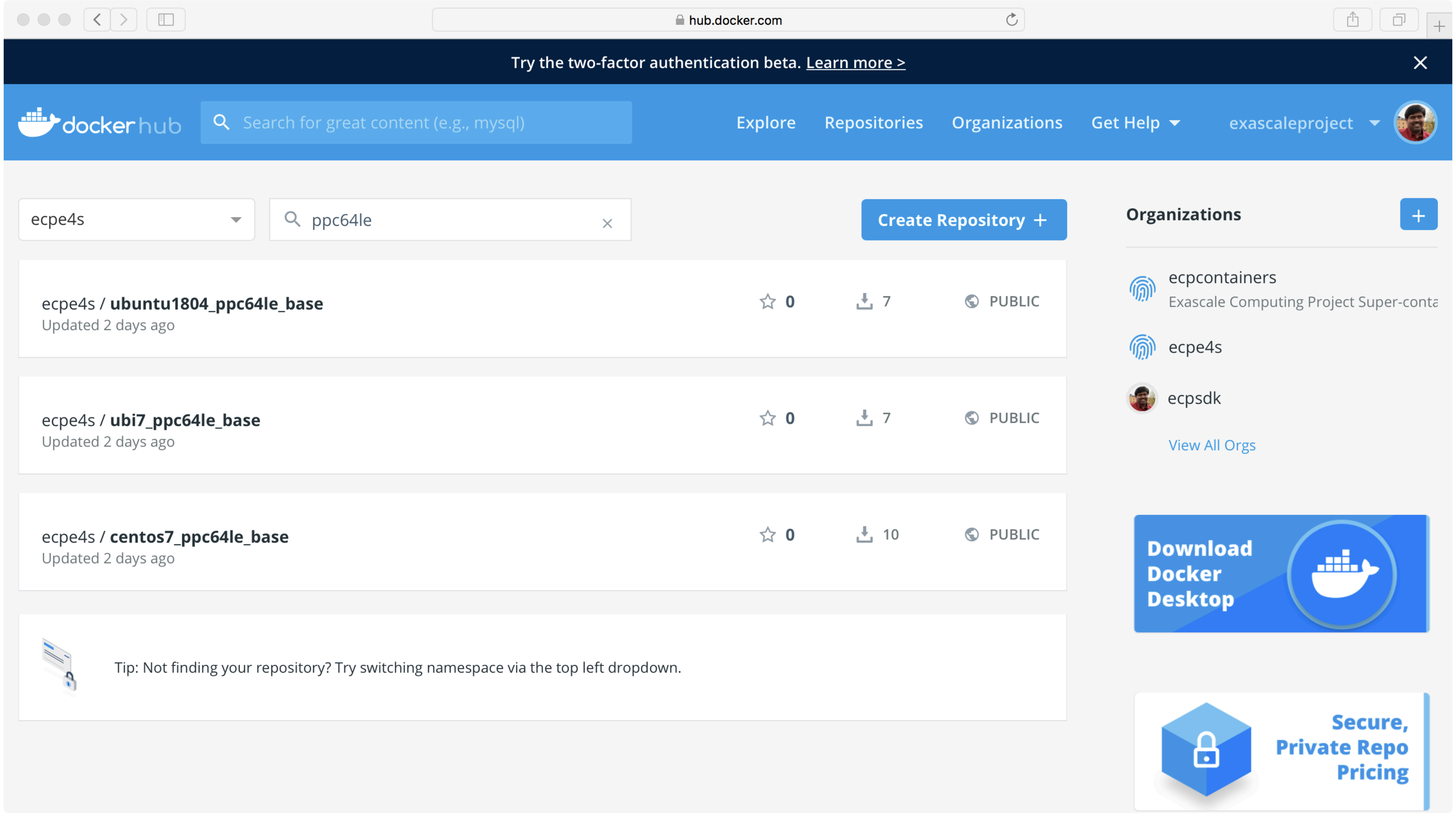
- wget http://oaciss.uoregon.edu/e4s/images/ubuntu18.04-e4s-gpu-ppc64le_1.2.simg
- singularity exec --nv ubuntu18.04-e4s-gpu-ppc64le_1.2.simg /bin/bash --rcfile /etc/bashrc
- spack find; module avail

E4S v2021-02 GPU Support

```
alias runsi='singularity exec --nv /home/users/sameer/images/ubuntu18.04-e4s-gpu-ppc64le_1.2.simg /bin/bash --rcfile /etc/bashrc'
[sameer@gorgon ~]$ runsi
Singularity> python
Python 3.6.10 |Anaconda, Inc.| (default, Jan  7 2020, 21:47:07)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow
>>> import torch
>>> import cv2
>>> import matplotlib
>>> import numpy
>>> tensorflow.test.is_gpu_available()
2020-11-05 17:09:35.705979: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1433] Found device 0 with properties:
name: Tesla V100-SXM2-32GB major: 7 minor: 0 memoryClockRate(GHz): 1.53
pciBusID: 0004:04:00.0
totalMemory: 31.75GiB freeMemory: 12.35GiB
2020-11-05 17:09:35.778351: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1433] Found device 1 with properties:
name: Tesla V100-SXM2-32GB major: 7 minor: 0 memoryClockRate(GHz): 1.53
pciBusID: 0004:05:00.0
totalMemory: 31.75GiB freeMemory: 31.44GiB
2020-11-05 17:09:35.907371: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1433] Found device 2 with properties:
name: Tesla V100-SXM2-32GB major: 7 minor: 0 memoryClockRate(GHz): 1.53
pciBusID: 0035:03:00.0
totalMemory: 31.75GiB freeMemory: 883.50MiB
2020-11-05 17:09:35.989499: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1433] Found device 3 with properties:
name: Tesla V100-SXM2-32GB major: 7 minor: 0 memoryClockRate(GHz): 1.53
pciBusID: 0035:04:00.0
totalMemory: 31.75GiB freeMemory: 31.44GiB
2020-11-05 17:09:35.989594: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1512] Adding visible gpu devices: 0, 1, 2, 3
2020-11-05 17:09:45.948104: I tensorflow/core/common_runtime/gpu/gpu_device.cc:984] Device interconnect StreamExecutor with strength 1 ed
ge matrix:
2020-11-05 17:09:45.948182: I tensorflow/core/common_runtime/gpu/gpu_device.cc:990]      0 1 2 3
2020-11-05 17:09:45.948199: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1003] 0:   N Y Y Y
2020-11-05 17:09:45.948210: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1003] 1:   Y N Y Y
2020-11-05 17:09:45.948222: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1003] 2:   Y Y N Y
2020-11-05 17:09:45.948232: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1003] 3:   Y Y Y N
2020-11-05 17:09:45.950552: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device (/device:GPU:0 with 11587
MB Snapz Pro X physical GPU (device: 0, name: Tesla V100-SXM2-32GB, pci bus id: 0004:04:00.0, compute capability: 7.0)
```

EXASCALE COMPUTING PROJECT

E4S: ppc64le Base Container Images

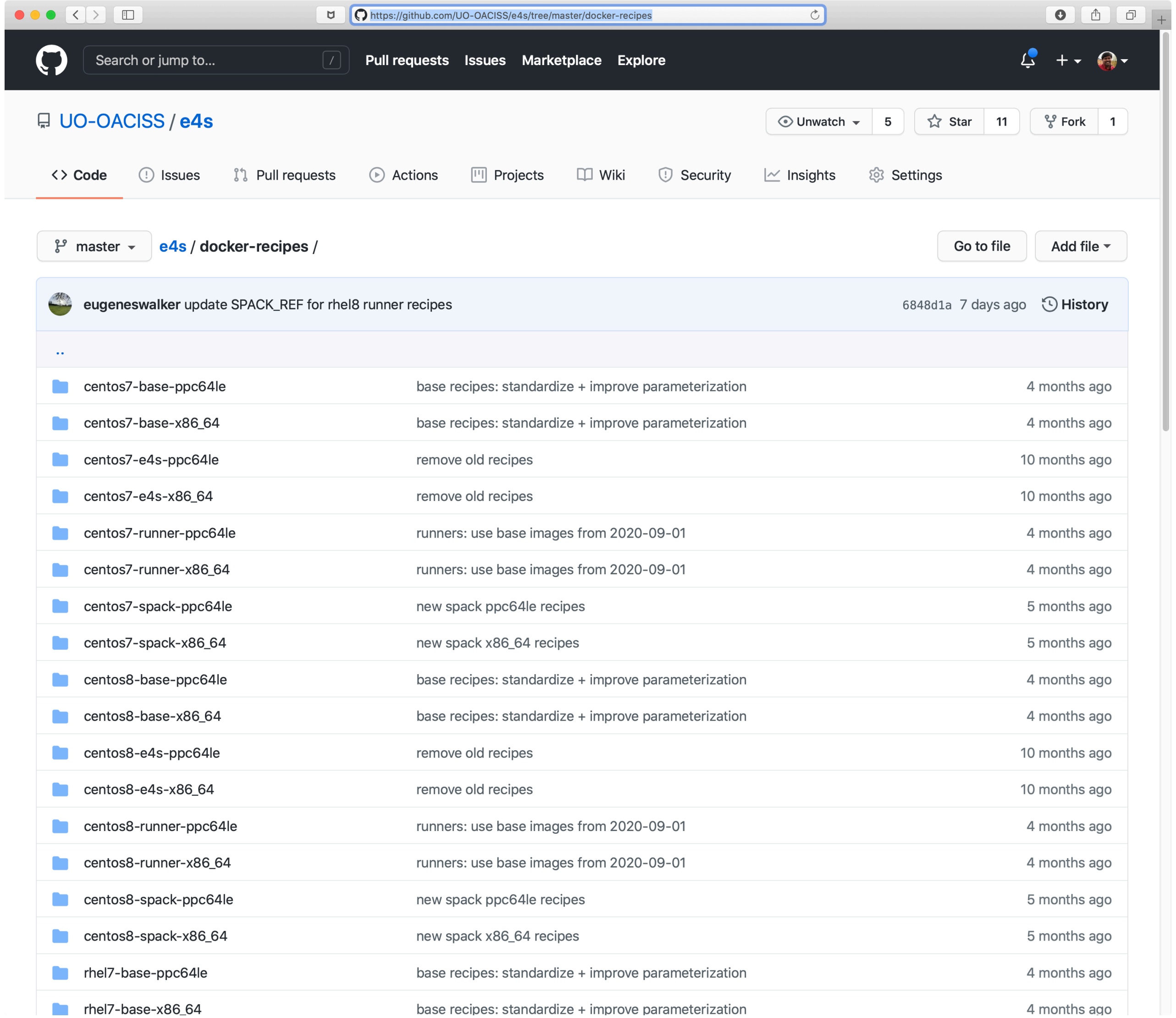


- Hub.docker.com
- ecpe4s

- Ubuntu 18.04
- RHEL/UBI 7.6
- Centos 7.6

EXASCALE COMPUTING PROJECT

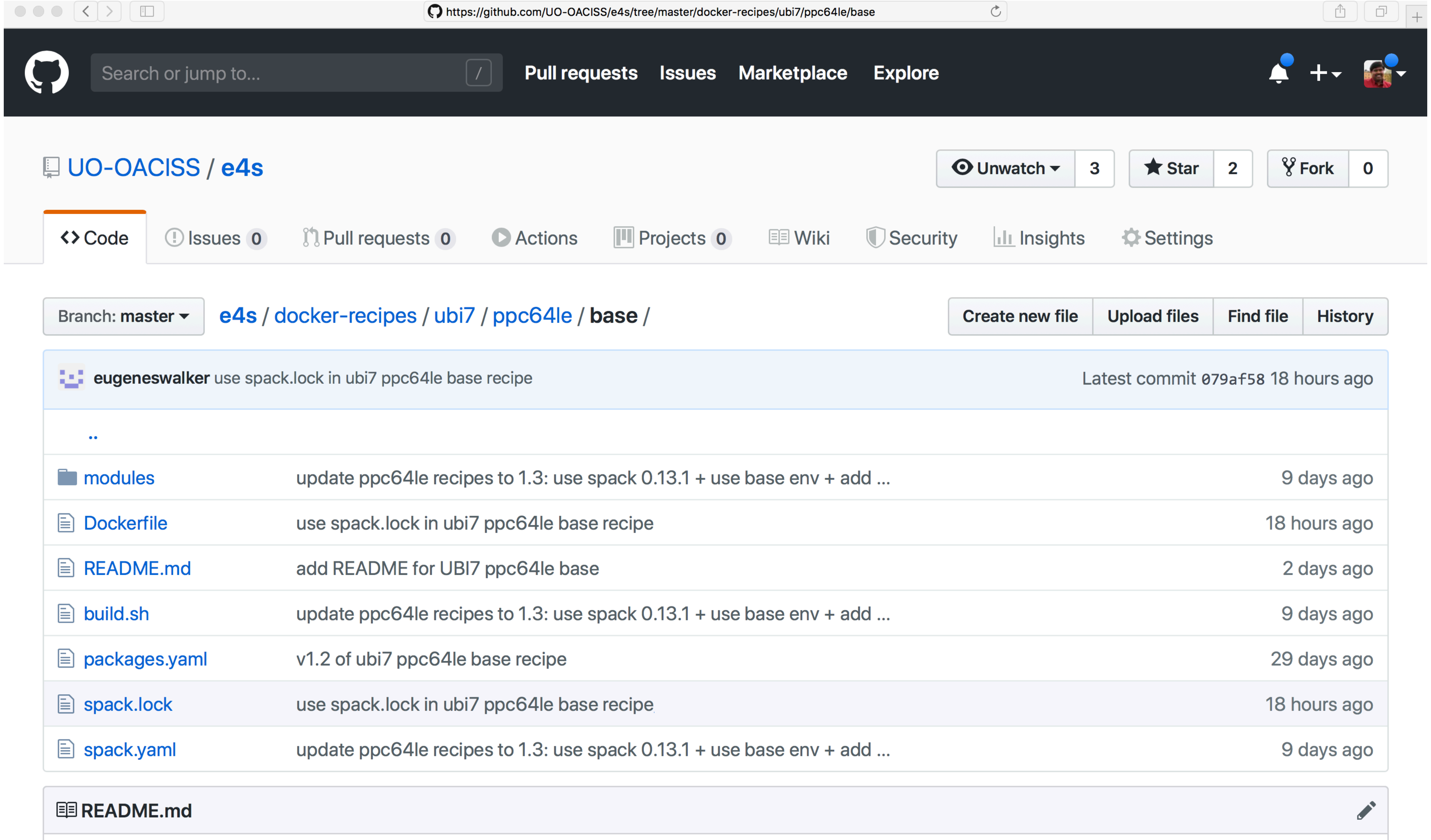
Multi-platform E4S Docker Recipes



10 lines (6 sloc) | 178 Bytes

```
FROM ecpe4s/ubuntu18.04-spack-x86_64:0.14.1

WORKDIR /e4s-env

COPY /spack.yaml .

RUN spack install --cache-only \
 && spack clean -a && rm -rf /tmp/root/spack-stage

WORKDIR /
```

E4S: Multi-platform Reproducible Docker Recipes



E4S
- x86_64
- ppc64le
- aarch64

https://e4s.io

E4S Spack environment spack.yaml

https://github.com/UO-OACISS/e4s/blob/master/docker-recipes/ubuntu18.04-e4s-x86_64/spack

Search or jump to... | Pull requests | Issues | Marketplace | Explore

UO-OACISS / e4s

Unwatch 5 | Star 11 | Fork 1

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

master | e4s / docker-recipes / ubuntu18.04-e4s-x86_64 / spack.yaml | Go to file

eugeneswalker remove old recipes — Latest commit 096d2ec on Mar 23, 2020 — History

1 contributor

201 lines (201 sloc) | 3.04 KB — Raw | Blame

```yaml
spack:
  concretization: separately
  definitions:
  - e4s:
    - openpmd-api
    - slate
    - magma
    - stc
    - py-libensemble^python@3.7.3
    - flecsi+cinch
    - hypre
    - mfem
    - trilinos@12.14.1+dtk+intrepid2+shards
    - sundials
    - petsc
    - strumpack
    - superlu-dist
    - superlu
    - tasmanian
    - openblas threads=openmp
    - faodel
    - mercury
    - hdf5
    - adios2
    - adios
    - darshan-runtime
    - darshan-util
    - veloc
    - scr
```

- Bare-metal install
  % cat spack.yaml
  % spack -e . install

- Docker build:

Executable File | 2 lines (2 sloc) | 78 Bytes

```bash
#!/bin/bash -x
docker build --no-cache -t ecpe4s/ubuntu18.04-e4s-x86_64:1.2 .
```

EXASCALE COMPUTING PROJECT

E4S: Spack Build Cache at U. Oregon

https://oaciss.uoregon.edu/e4s/inventory.html

E4S Build Cache for Spack 0.16.1

To use this build cache, just add it to your Spack

spack mirror add E4S https://cache.e4s.io

wget https://oaciss.uoregon.edu/e4s/e4s.pub

spack gpg trust e4s.pub

Click on one of the packages below to see a list of all available variants.

All Architectures PPC64LE X86_64

All Operating Systems Centos 7 Centos 8 RHEL 7 RHEL 8 Ubuntu 18.04 Ubuntu 20.04

Last updated: 03-15-2021 20:55 PDT

37869 Spack packages

Search

adiak@0.1.1 adiak@0.2.1 adios2@2.5.0 adios2@2.6.0 adios2@2.7.0 adios2@2.7.1 adios@1.13.1 adlbx@0.9.2 adol-c@2.7.2 amg@1.2 aml@0.1.0 amrex@20.07 amrex@20.09 amrex@20.10 amrex@20.11 amrex@20.12 amrex@21.01 amrex@21.02

amrex@21.03

Click on the full spec link to find out more.

Link	Arch	OS	Compiler	Created	Full Hash
Full Spec	ppc64le	rhel8	gcc@8.3.1	03-02-2021 08:14 PST	aoumvudyk5supalidurl72xgyow6odr6
Full Spec	ppc64le	rhel8	gcc@8.3.1	03-11-2021 22:21 PST	mz37n35i5dvrwd3dy5wtxo3vybmnw5ma
Full Spec	ppc64le	ubuntu18.04	gcc@7.5.0	03-02-2021 08:24 PST	xumh5lzbuhigavi34mmuopuyrzz7ifw7
Full Spec	ppc64le	ubuntu18.04	gcc@7.5.0	03-11-2021 10:52 PST	wdlmiskn6paelpwwzpd67uv44nn334wi
Full Spec	ppc64le	ubuntu20.04	gcc@9.3.0	03-02-2021 08:24 PST	undav3wfl7rwmsivzdpohw5otpdnrhwp
Full Spec	ppc64le	ubuntu20.04	gcc@9.3.0	03-11-2021 14:45 PST	irrw5ysyikvbtg3rwfkxiycqjehwcbhp
Full Spec	x86_64	rhel8	gcc@8.3.1	03-04-2021 10:47 PST	gbwdkpn7vzyklwim4noaifmpptl2okfe
Full Spec	x86_64	rhel8	gcc@8.3.1	03-11-2021 21:41 PST	uddagvkiyjbtrgpttodq5uxv76gs5e5h
Full Spec	x86_64	ubuntu18.04	gcc@7.5.0	03-02-2021 08:06 PST	wdg7eczy7nwfj2lqb7yc7qjzyhtt5f7v
Full Spec	x86_64	ubuntu18.04	gcc@7.5.0	03-11-2021 10:28 PST	gfebiwzie32j7xx7qg77szzhlj5b6woa
Full Spec	x86_64	ubuntu20.04	gcc@9.3.0	03-02-2021 08:06 PST	vcpmk5oo4t2xxolinkk75vn3uxkxiucc
Full Spec	x86_64	ubuntu20.04	gcc@9.3.0	03-11-2021 14:25 PST	c5634irwnlkmmxpiq32wirbsxtosqlcg

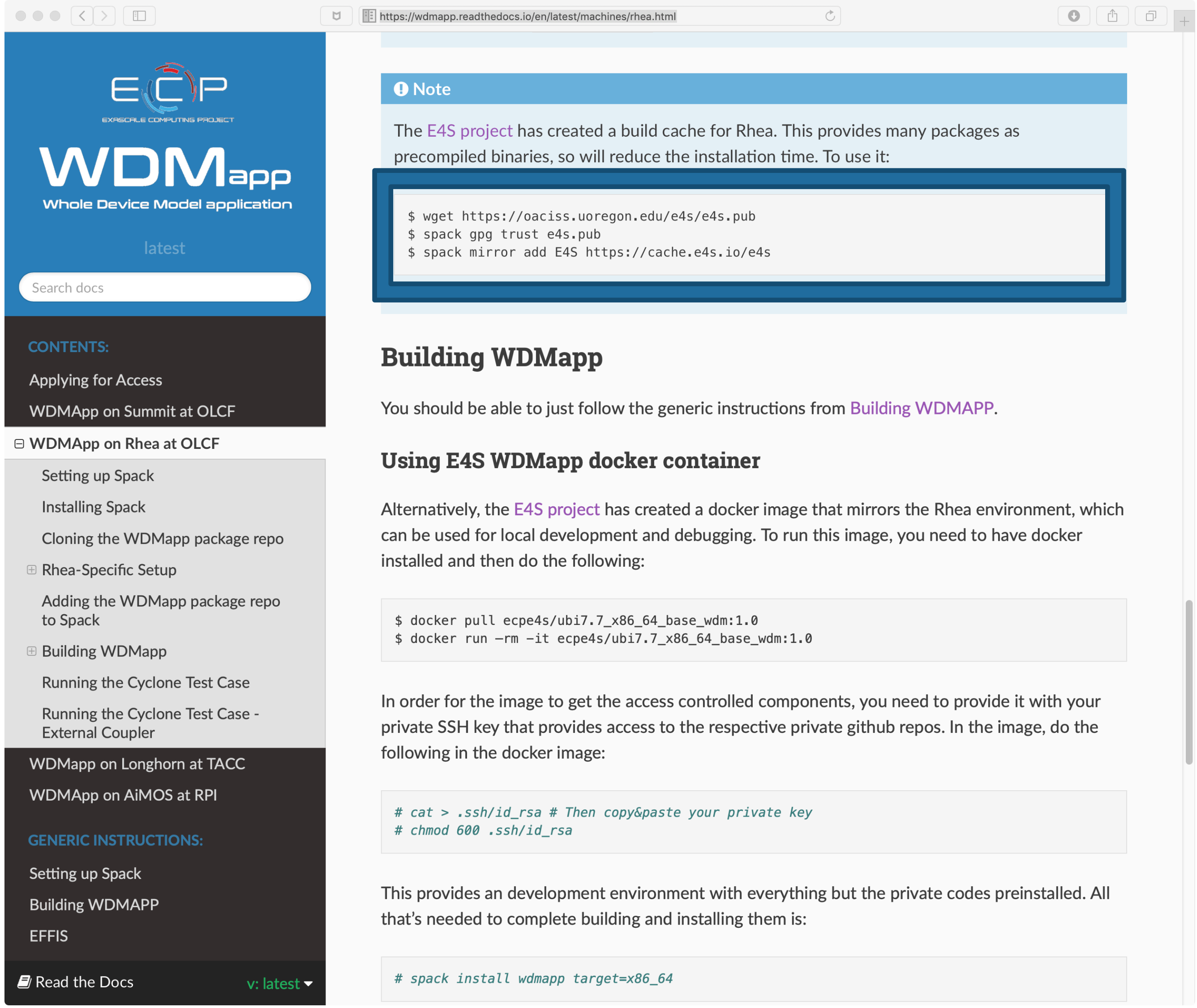
ant@1.10.0 ant@1.10.7 arborx@0.9-beta argobots@1.0 argobots@1.0rc1 argobots@1.0rc2 arpack-ng@3.7.0 arpack-ng@3.8.0 ascent@0.6.0 ascent@develop assimp@4.0.1

- 37,000+ binaries
- S3 mirror
- No need to build from source code!

- https://oaciss.uoregon.edu/e4s/inventory.html

EXASCALE COMPUTING PROJECT

WDMApp: Speeding up bare-metal installs using E4S build cache


https://wdmapp.readthedocs.io/en/latest/machines/rhea.html

WDMapp
Whole Device Model application
latest

Search docs

CONTENTS:

GENERIC INSTRUCTIONS:

Read the Docs v: latest

Note

The E4S project has created a build cache for Rhea. This provides many packages as precompiled binaries, so will reduce the installation time. To use it:

```
$ wget https://oaciss.uoregon.edu/e4s/e4s.pub
$ spack gpg trust e4s.pub
$ spack mirror add E4S https://cache.e4s.io/e4s
```

Building WDMapp

You should be able to just follow the generic instructions from Building WDMAPP.

Using E4S WDMapp docker container

Alternatively, the E4S project has created a docker image that mirrors the Rhea environment, which can be used for local development and debugging. To run this image, you need to have docker installed and then do the following:

```
$ docker pull ecpe4s/ubi7.7_x86_64_base_wdm:1.0
$ docker run —rm —it ecpe4s/ubi7.7_x86_64_base_wdm:1.0
```

In order for the image to get the access controlled components, you need to provide it with your private SSH key that provides access to the respective private github repos. In the image, do the following in the docker image:

```
# cat > .ssh/id_rsa # Then copy&paste your private key
# chmod 600 .ssh/id_rsa
```

This provides an development environment with everything but the private codes preinstalled. All that's needed to complete building and installing them is:

```
# spack install wdmapp target=x86_64
```

- E4S Spack build cache
- Adding E4S mirror
- WDMApp install speeds up!

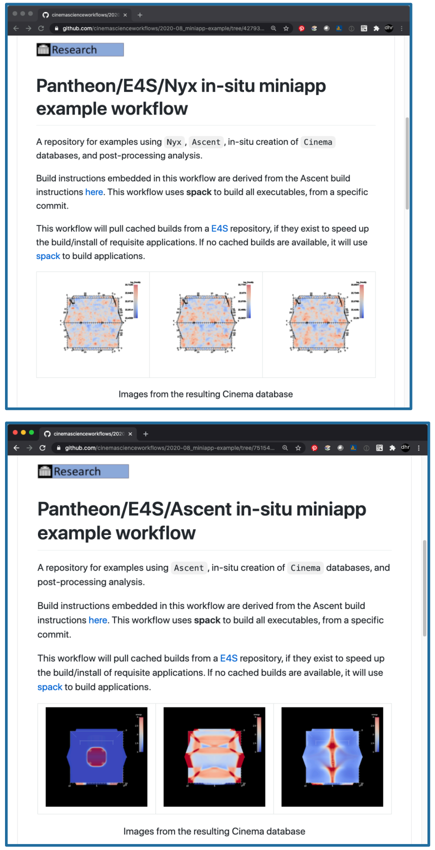
Pantheon and E4S build cache support end-to-end ECP examples

Overview: The Exascale Computing Project (ECP) is a complex undertaking, involving a myriad of technologies working together. An outstanding need is a way to capture, curate, communicate and validate workflows that cross all of these boundaries.

The **Pantheon** and **E4S** projects are collaborating to advance the integration and testing of capabilities, and to promote understanding of the complex workflows required by the ECP project. Utilizing a host of ECP technologies (spack, Ascent, Cinema, among others), this collaboration brings curated workflows to the fingertips of ECP researchers.

Contributions

- Curated end-to-end application/in-situ analysis examples can be run quickly by anyone on Summit. (https://github.com/pantheonscience/ECP-E4S-Examples)
- Pantheon/E4S integration speeds up build/setup times over source builds due to cached binaries (approx.10x speed up).



Instructions page for (top) Nyx, Ascent and Cinema workflow repository, and (bottom) Cloverleaf3d, Ascent, Cinema workflow. These curated workflows use Pantheon, E4S and spack to provide curated workflows for ECP.

LA-UR-20-27327 4/11/21

EXASCALE COMPUTING PROJECT

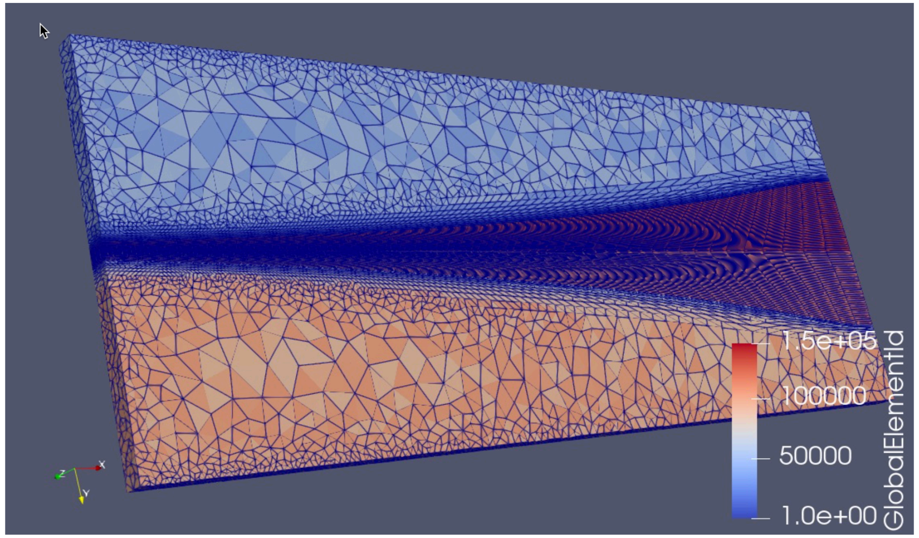
PANTHEON

Nalu-Wind Demo

ExaWind Project

Nalu-Wind

- Wind-specific version of the Nalu code
- developed at SNL
- https://github.com/exawind/nalu-wind
- Incompressible-flow computational fluid
- dynamics (CFD) code
- Unstructured-grid finite-volume
- discretization
- https://www.github.com/Exawind



EXASCALE COMPUTING PROJECT

NREL
NATIONAL RENEWABLE ENERGY LABORATORY

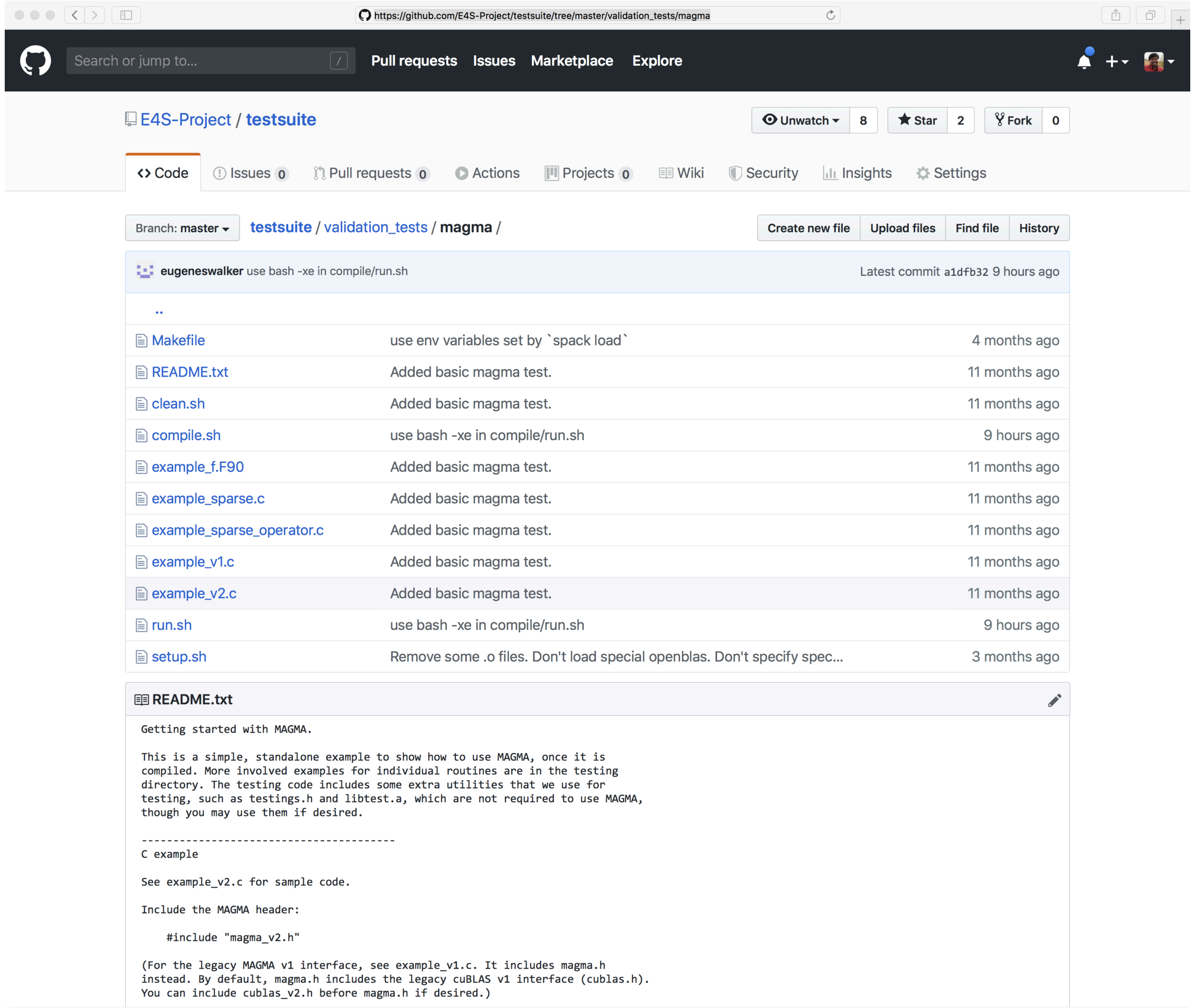
Sandia National Laboratories

OAK RIDGE National Laboratory

TEXAS
The University of Texas at Austin

E4S Validation Test Suite

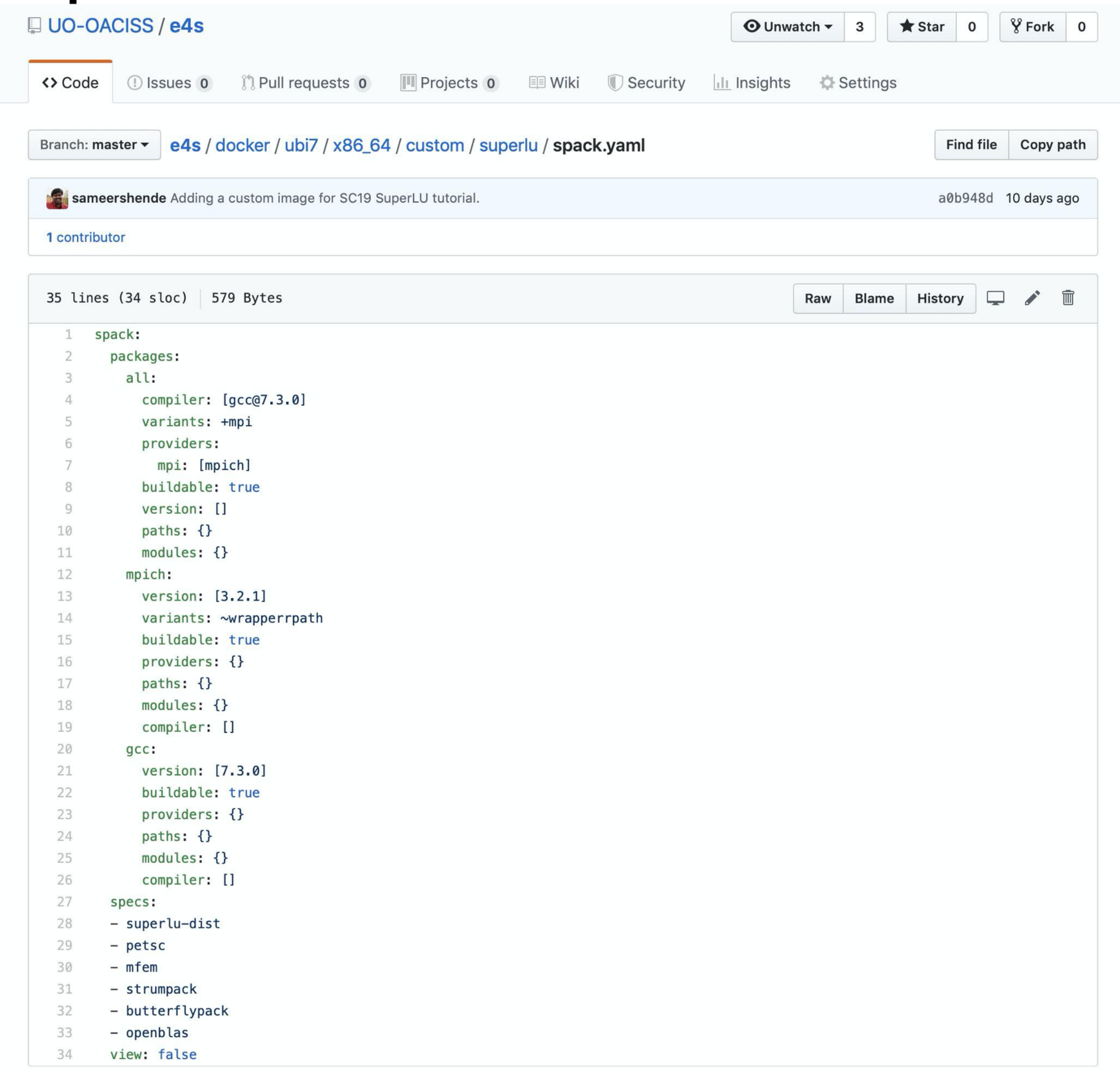
- Provides automated build and run tests
- Validate container environments and products
- New LLVM validation test suite for DOE LLVM



EXASCALE COMPUTING PROJECT

- git clone https://github.com/E4S-Project/testsuite.git

Reproducible Container Builds using E4S Base Images



UO-OACISS / e4s

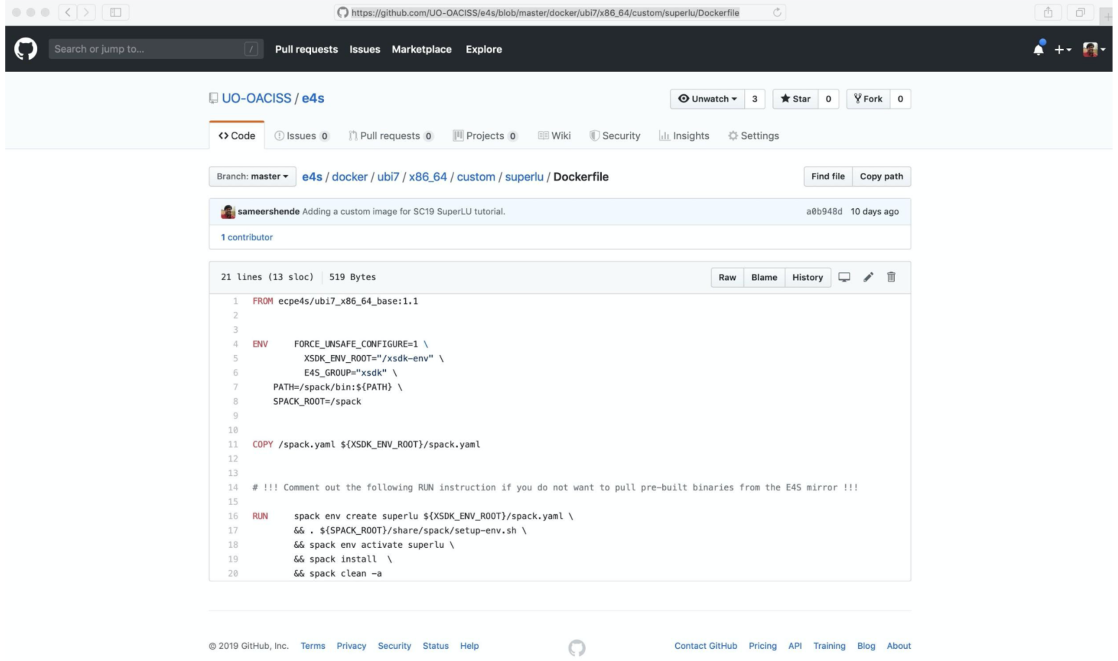
e4s / docker / ubi7 / x86_64 / custom / superlu / spack.yaml

sameershende Adding a custom image for SC19 SuperLU tutorial. a0b948d 10 days ago

1 contributor

35 lines (34 sloc) | 579 Bytes

```yaml
spack:
  packages:
    all:
      compiler: [gcc@7.3.0]
      variants: +mpi
      providers:
        mpi: [mpich]
      buildable: true
      version: []
      paths: {}
      modules: {}
    mpich:
      version: [3.2.1]
      variants: ~wrapperrpath
      buildable: true
      providers: {}
      paths: {}
      modules: {}
      compiler: []
    gcc:
      version: [7.3.0]
      buildable: true
      providers: {}
      paths: {}
      modules: {}
      compiler: []
  specs:
  - superlu-dist
  - petsc
  - mfem
  - strumpack
  - butterflypack
  - openblas
  view: false
```



UO-OACISS / e4s

e4s / docker / ubi7 / x86_64 / custom / superlu / Dockerfile

sameershende Adding a custom image for SC19 SuperLU tutorial. a0b948d 10 days ago

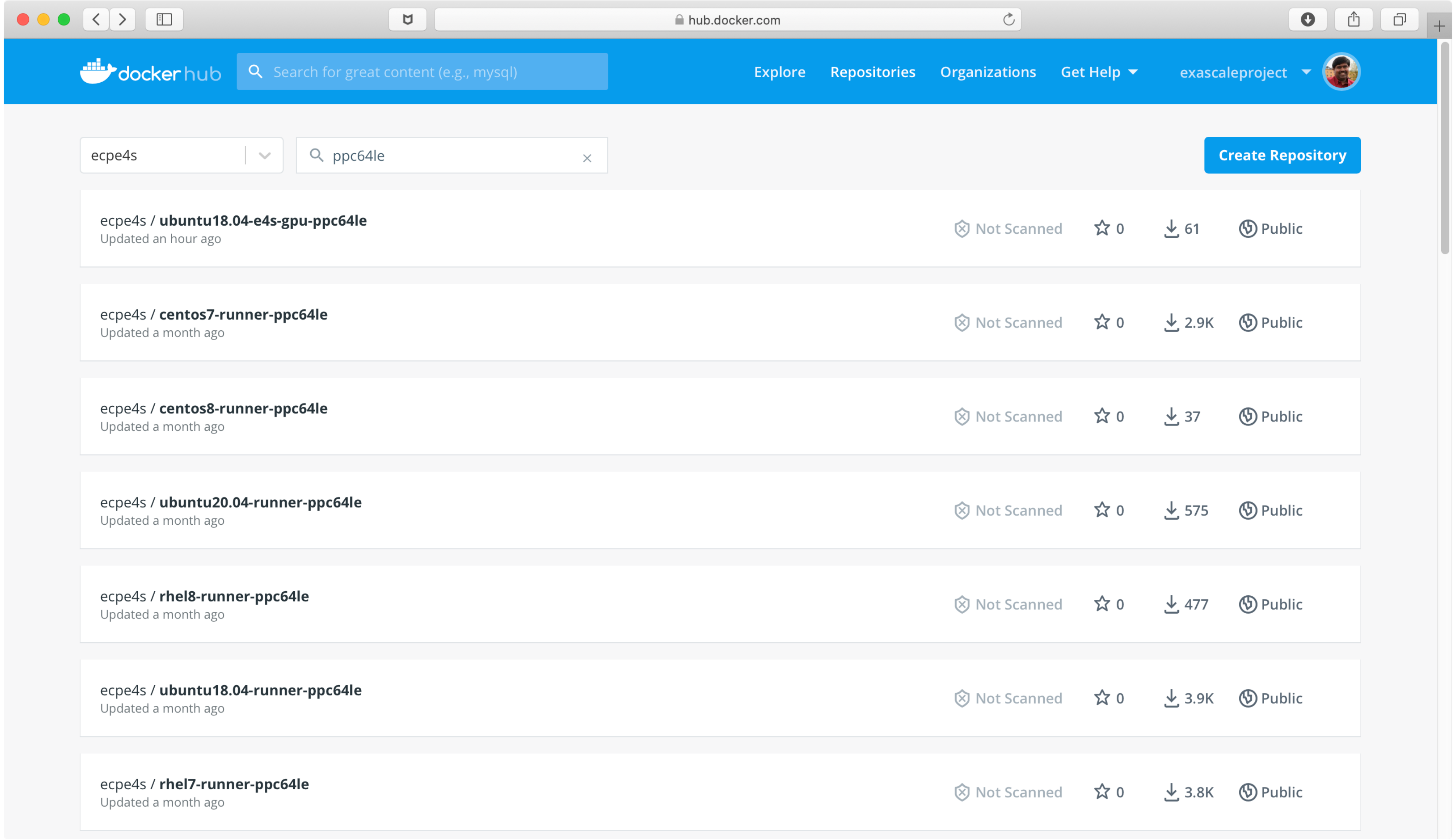
1 contributor

21 lines (13 sloc) | 519 Bytes

```dockerfile
FROM ecpe4s/ubi7_x86_64_base:1.1


ENV     FORCE_UNSAFE_CONFIGURE=1 \
          XSDK_ENV_ROOT="/xsdk-env" \
          E4S_GROUP="xsdk" \
    PATH=/spack/bin:${PATH} \
    SPACK_ROOT=/spack


COPY /spack.yaml ${XSDK_ENV_ROOT}/spack.yaml


# !!! Comment out the following RUN instruction if you do not want to pull pre-built binaries from the E4S mirror !!!

RUN     spack env create superlu ${XSDK_ENV_ROOT}/spack.yaml \
        && . ${SPACK_ROOT}/share/spack/setup-env.sh \
        && spack env activate superlu \
        && spack install  \
        && spack clean -a
```

- PMR SDK base image has Spack build cache mirror and GPG key installed.
- Base image has GCC and MPICH configured for MPICH ABI level replacement (with system MPI).
- Customized container build using binaries from E4S Spack build cache for fast deployment.
- No need to rebuild packages from the source code.
- Same recipe for container and native bare-metal builds with Spack!

E4S: GitLab Runner Images



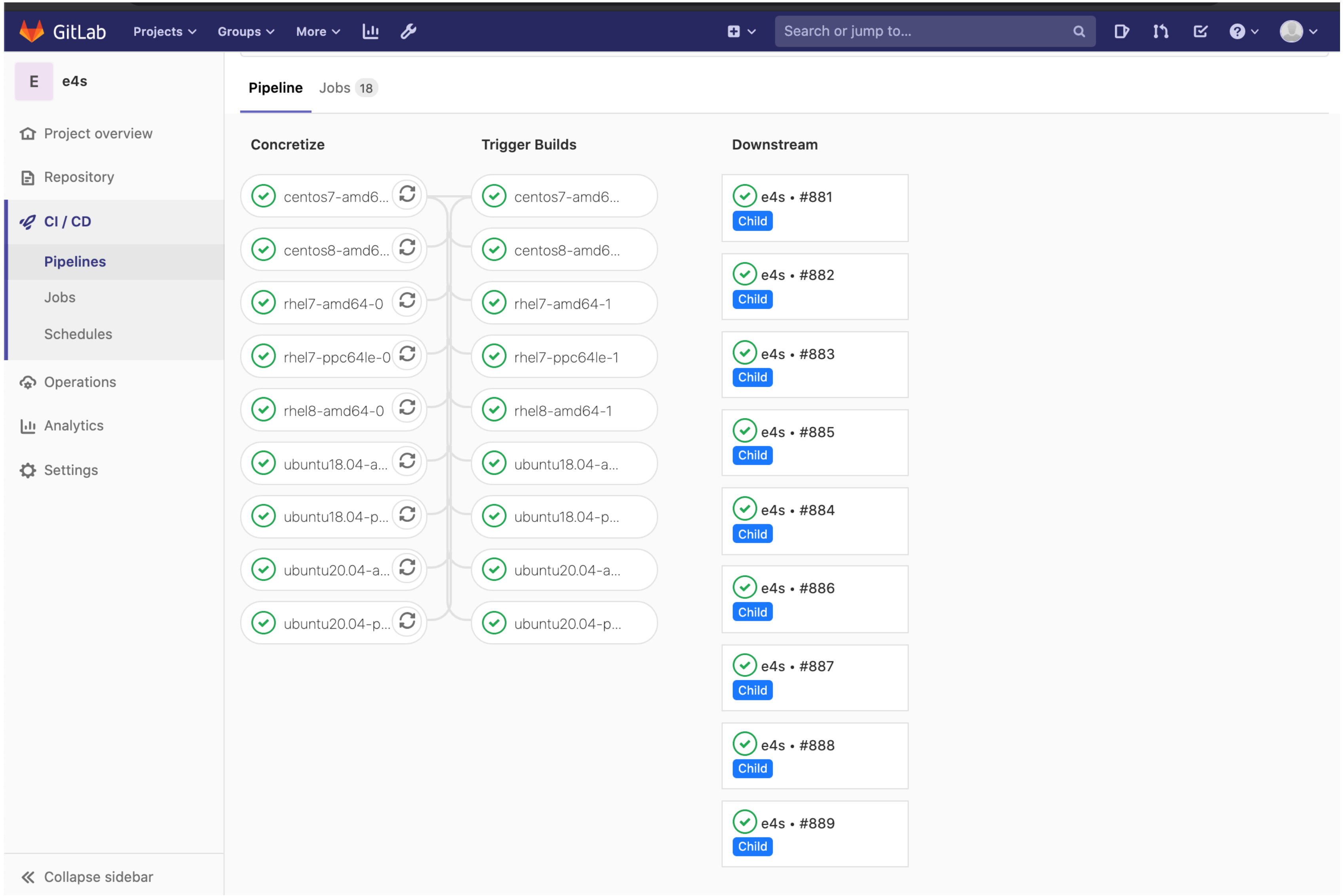
- Dockerhub
- Bare-bones
- Multi-platfrom
- Build E4S

EXASCALE COMPUTING PROJECT

University of Oregon GitLab CI

E4S Builds:
•Ubuntu 18.04
•Ubuntu 20.04
•RHEL 7.6
•RHEL 8
•CentOS 7
•CentOS 8

Architectures:
ppc64le and x86_64



- https://gitlab.e4s.io

GitLab GPU Runners on Frank, U. Oregon

Recent searches | uo-illyad | Created date

Runners currently online: 11

Type/State	Runner token	Description	Version	IP Address	Projects	Jobs	Tags	Last contact
shared locked paused	9yUY6sjq	uo-illyad	13.9.0	192.168.165.1...	n/a	0	a100 vx avx2 arge medium nvidia-gpu public small spack uo x86_64 xlarge	3 minutes ago

A100 NVIDIA GPU

Recent searches | uo-jupiter | Created date

Runners currently online: 11

Type/State	Runner token	Description	Version	IP Address	Projects	Jobs	Tags	Last contact
specific locked paused	E_zdx2ry	uo-jupiter	13.9.0	192.168.165.1...	2	105	avx avx2 cooper-lal dg1 huge int large medium public small spack uo x86_64 xlarge	4 minutes ago

DG1 Intel GPU

Recent searches | uo-instinct | Created date

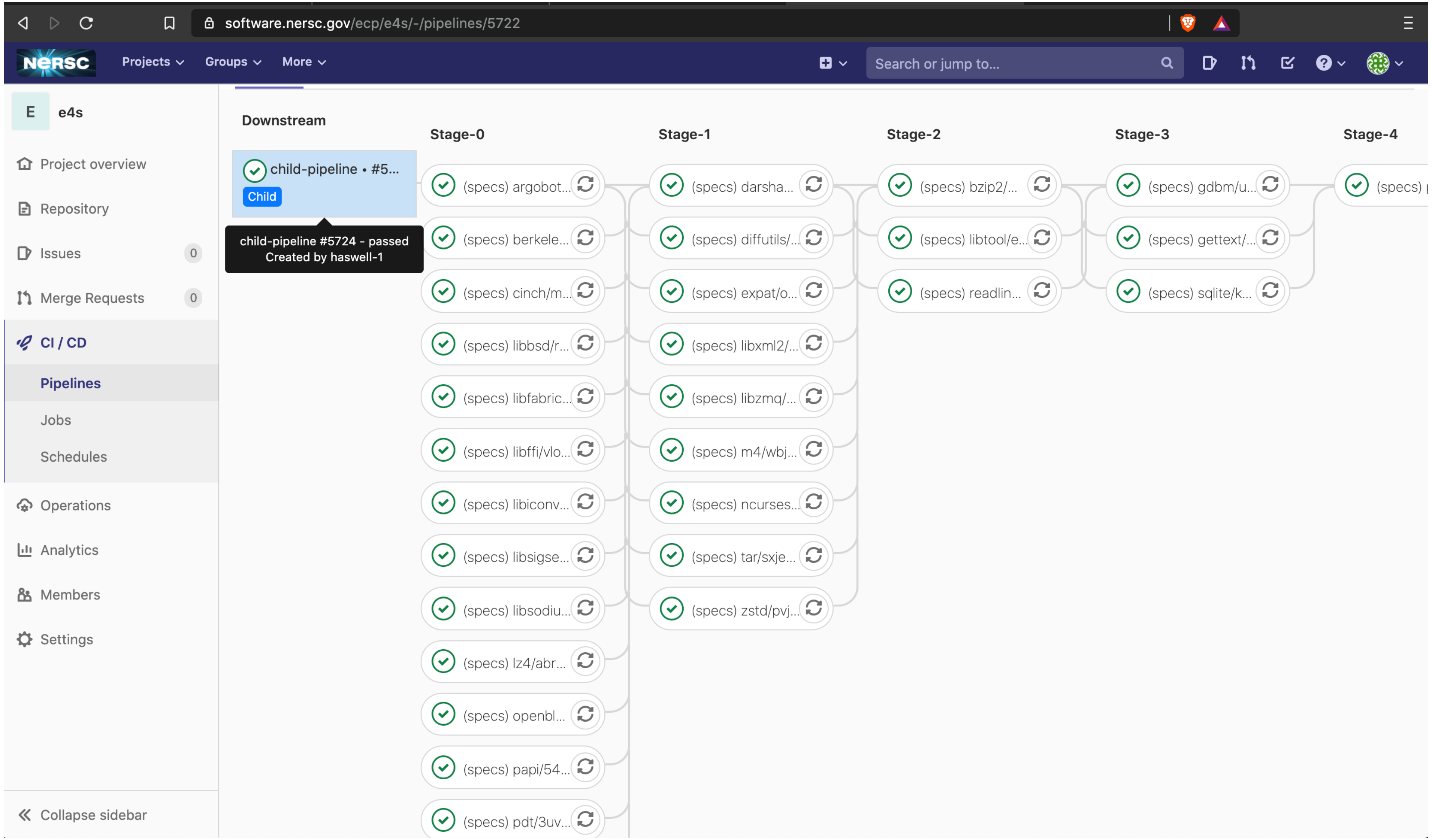
Runners currently online: 11

Type/State	Runner token	Description	Version	IP Address	Projects	Jobs	Tags	Last contact
shared locked paused	xPVjy9oY	uo-instinct	13.9.0	192.168.165.1...	n/a	0	amd-gpu avx avx2 mediur mi50 public spack uo x86_64 xlarge	5 minutes ago

MI50 AMD GPU

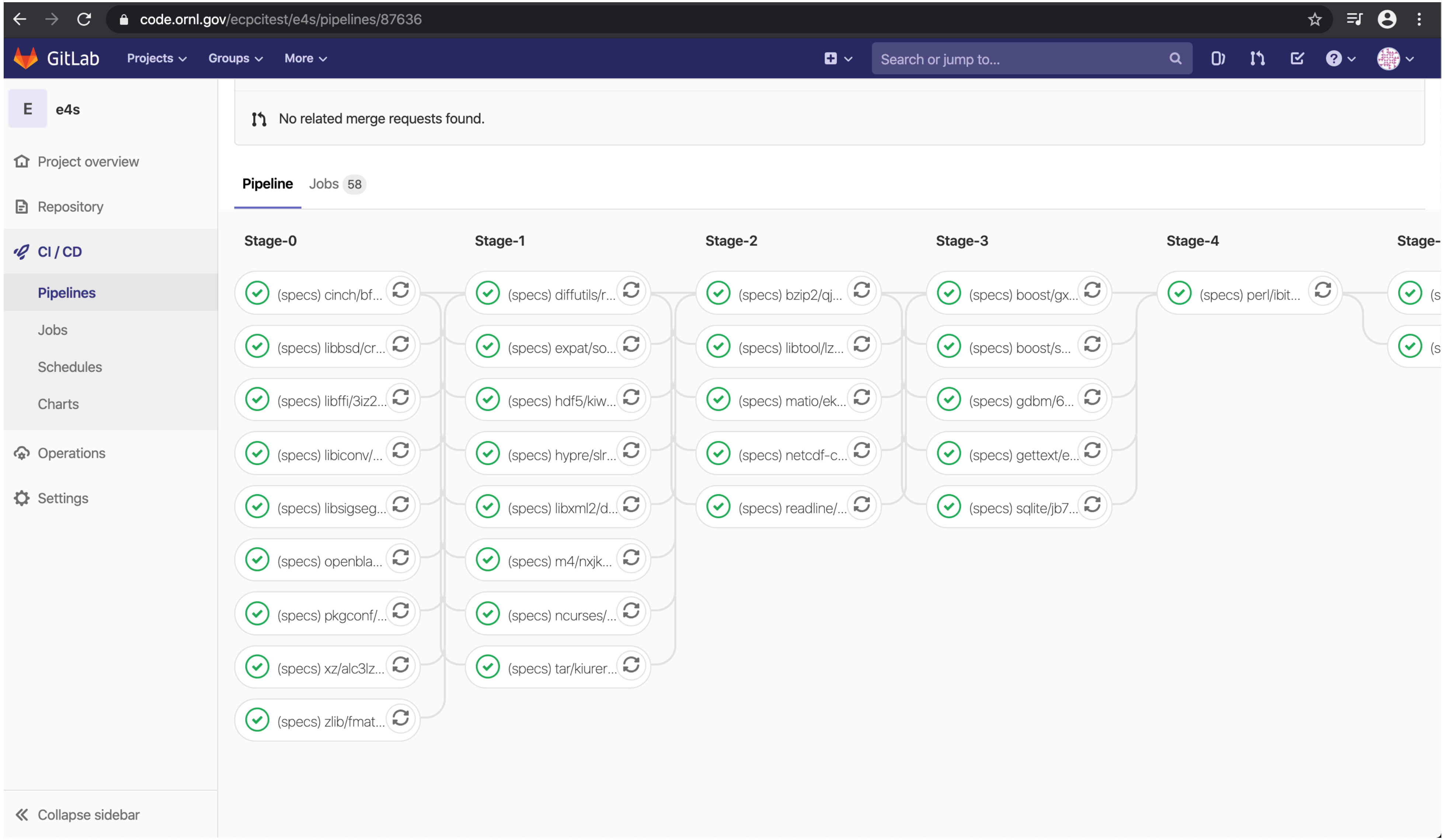
EXASCALE COMPUTING PROJECT

Multi-stage E4S CI Build Pipeline on Cori, NERSC



EXASCALE COMPUTING PROJECT

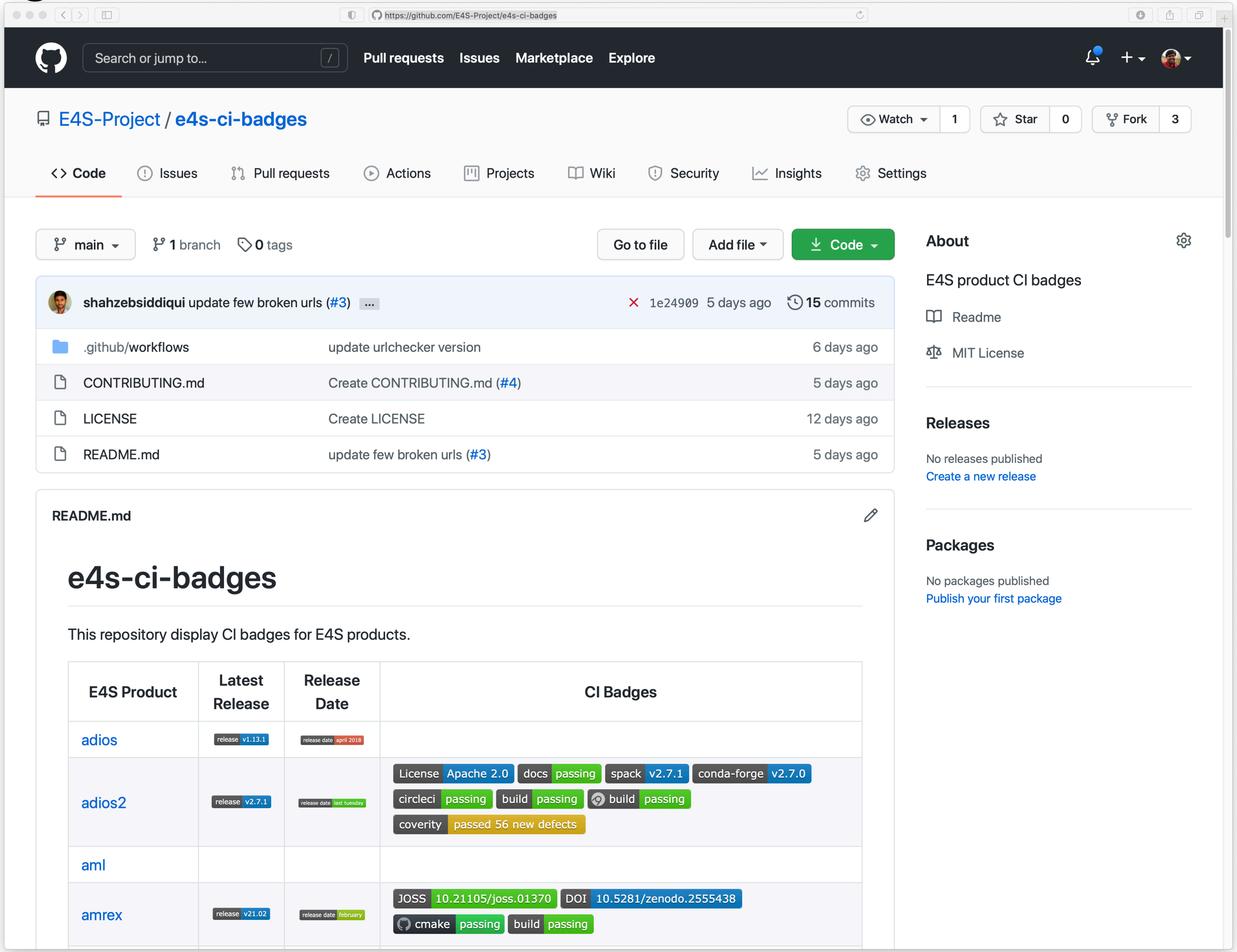
ORNL GitLab Build Pipeline for E4S Spack Build Cache



- ppc64le (Ascent @ ORNL)
- Reproducible container builds

EXASCALE COMPUTING PROJECT

E4S CI Badges

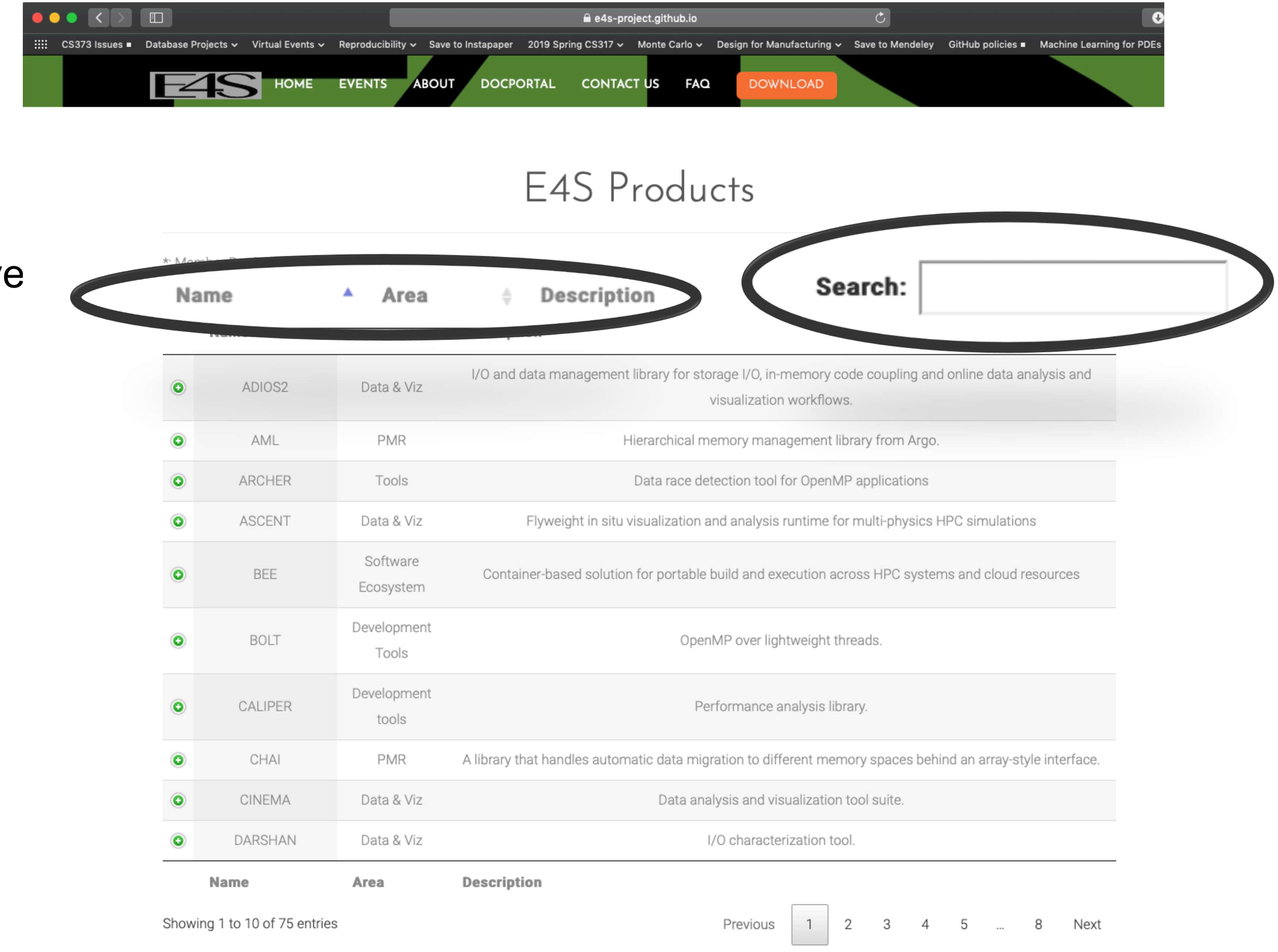


E4S DocPortal

- Provide a single online location for *accurate* product descriptions for ECP software products.
- Derived requirements:
  - Sustainable: Must be integrated into software team workflows.
  - Incremental: Must build on community approaches to providing this kind of information.
  - Extensible: Must be usable by any open source software team.
- Strategy:
  - Use the open source community approach of specially-name files in software repositories.
  - Adopt commonly used file names when available.
  - Identify new information items not already being requested.
  - Develop new special file names for information beyond what is already captured.
  - Create web-based raking tool to capture information from product repositories and present in summary form on a webpage.
  - Aggregates and summarizes documentation and metadata for E4S products
  - Regularly updates information directly from product repositories
  - Prototype: https://e4s-project.github.io/DocPortal.html

E4S DocPortal

- The DocPortal is live
- Summary Info
  - Name
  - Functional Area
  - Description
  - License
- Searchable
- Sortable



Name	Area	Description
ADIOS2	Data & Viz	I/O and data management library for storage I/O, in-memory code coupling and online data analysis and visualization workflows.
AML	PMR	Hierarchical memory management library from Argo.
ARCHER	Tools	Data race detection tool for OpenMP applications
ASCENT	Data & Viz	Flyweight in situ visualization and analysis runtime for multi-physics HPC simulations
BEE	Software Ecosystem	Container-based solution for portable build and execution across HPC systems and cloud resources
BOLT	Development Tools	OpenMP over lightweight threads.
CALIPER	Development tools	Performance analysis library.
CHAI	PMR	A library that handles automatic data migration to different memory spaces behind an array-style interface.
CINEMA	Data & Viz	Data analysis and visualization tool suite.
DARSHAN	Data & Viz	I/O characterization tool.

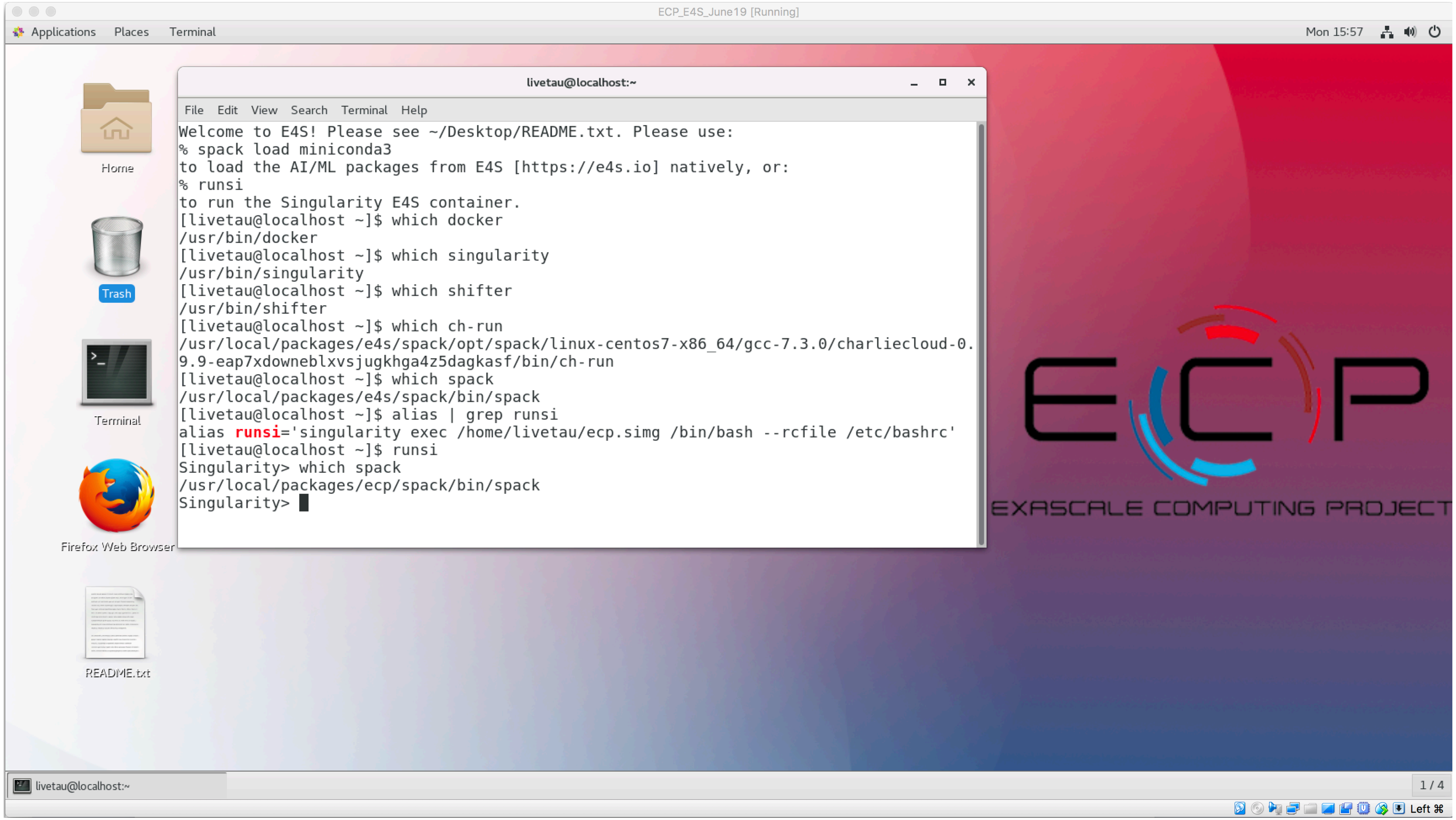
Showing 1 to 10 of 75 entries

Previous 1 2 3 4 5 ... 8 Next

https://e4s-project.github.io/DocPortal.html

EXASCALE COMPUTING PROJECT

E4S VirtualBox Image



Container Runtimes

- Docker
- Shifter
- Singularity
- Charliecloud

https://e4s.io

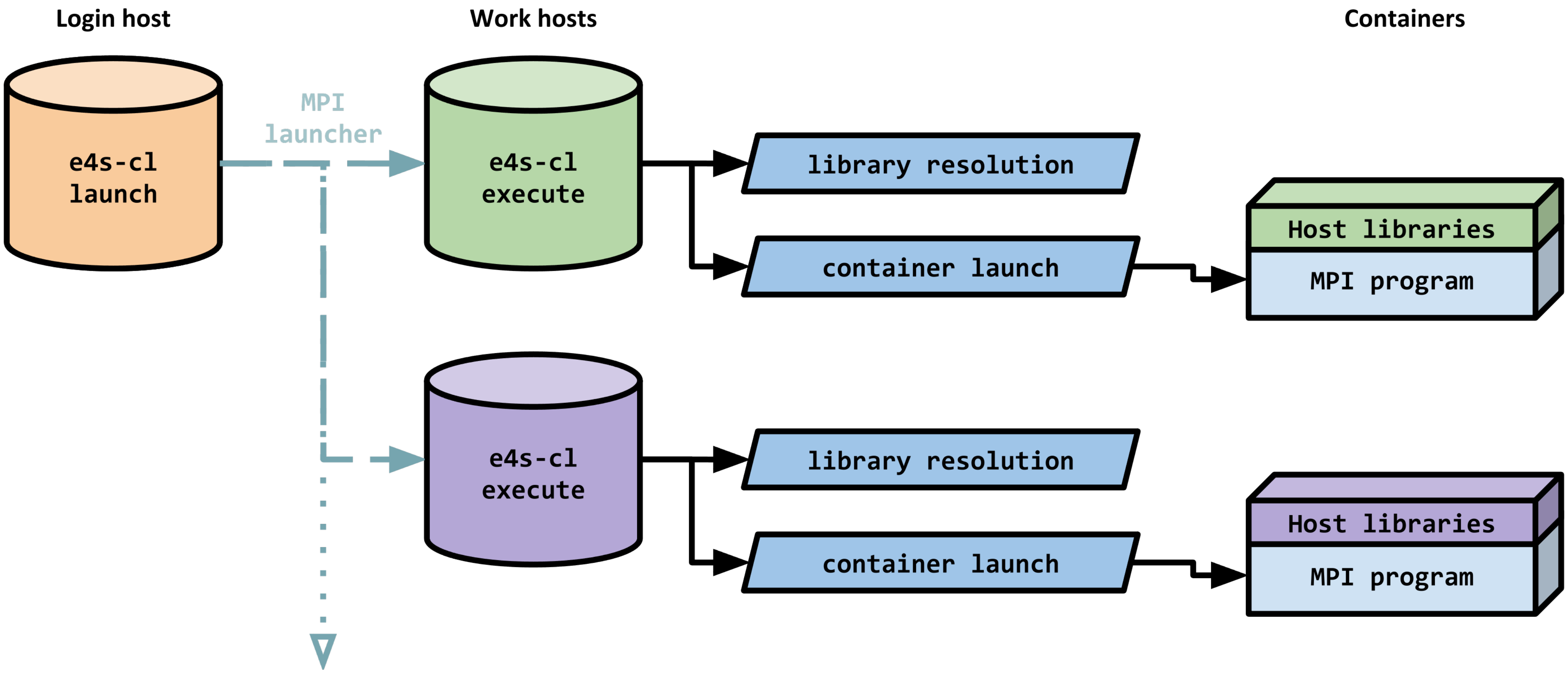
e4s-cl: A tool to simplify the launch of MPI jobs in E4S containers

- E4S containers support replacement of MPI libraries using MPICH ABI compatibility layer.
- 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.
- Usage:
  1. `e4s-cl init …`
  2. `e4s-cl mpirun -np <> -hosts <> <command>`

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

EXASCALE COMPUTING PROJECT

e4s-cl Container Launcher



E4S Summary

What E4S is not	• What E4S is
A closed system taking contributions only from DOE software development teams.	Extensible, open architecture software ecosystem accepting contributions from US and international teams. Framework for collaborative open-source product integration.
A monolithic, take-it-or-leave-it software behemoth.	A full collection of compatible software capabilities **and** A manifest of a la carte selectable software capabilities.
A commercial product.	Vehicle for delivering high-quality reusable software products in collaboration with others.
A simple packaging of existing software.	The conduit for future leading edge HPC software targeting scalable next-generation computing platforms. A hierarchical software framework to enhance (via SDKs) software interoperability and quality expectations.

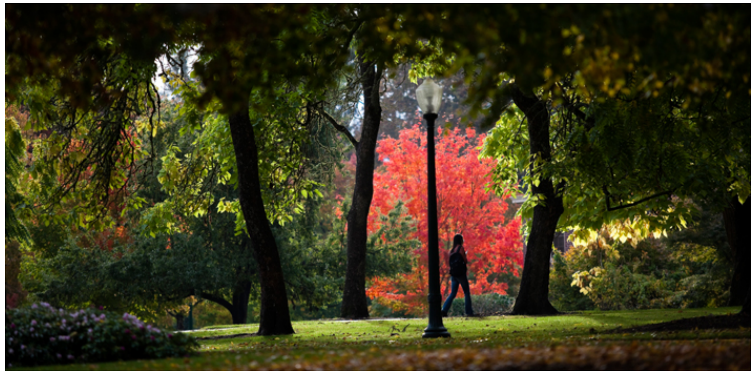
Future work, issues…

- Improved support for GPUs and visualization tools
- DOE LLVM
- Addition of CI testing
- Facility deployment
- Scalable startup with full-featured “Supercontainers”
- Improving the launch of MPI applications
- From-source builds assisted by a binary build cache or containers
- Docker and Singularity images are available for download
    - https://e4s.io

Vision for E4S Now and in the Future

- E4S has emerged as a new top-level component in the DOE HPC community, enabling fundamentally new relationships
- E4S has similar potential for new interactions with other US agencies, US industry and international collaborators. NSF and UK are examples
- The E4S portfolio can expand to include new domains (ML/AI), lower—level components (OS), and more.
- E4S can provide better (increased quality), faster (timely delivery of leading-edge capabilities) and cheaper (assisting product teams)

Performance Research Laboratory, University of Oregon, Eugene



www.uoregon.edu

Support Acknowledgements for TAU demo in Nalu-Wind

- US Department of Energy (DOE)
  - ANL
  - Office of Science contracts, ECP
  - SciDAC, LBL contracts
  - LLNL-LANL-SNL ASC/NNSA contract
  - Battelle, PNNL and ORNL contract
- Department of Defense (DoD)
  - PETTT, HPCMP
- National Science Foundation (NSF)
  - SI2-SSI, Glassbox
- NASA
- CEA, France
- AWS
- Partners:
  –University of Oregor
  –The Ohio State University
  –ParaTools, Inc.
  –University of Tennessee, Knoxville
  –T.U. Dresden, GWT
  –Jülich Supercomputing Center

Acknowledgment

"This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation's exascale computing imperative."

E4S Hands-on

Login to AWS instance and follow instructions in README file

- Reserve a unique host by entering your contact info at the Google Doc:
  - **https://e4s.io/tutorial**
- Use a web browser for DCV (remote desktop) software

  https://[hostname]:8443/#e4s

  (e.g., https://tut032.supercontainers.org:8443/#e4s

  if you reserved instance #32 or tut032.supercontainers.org)

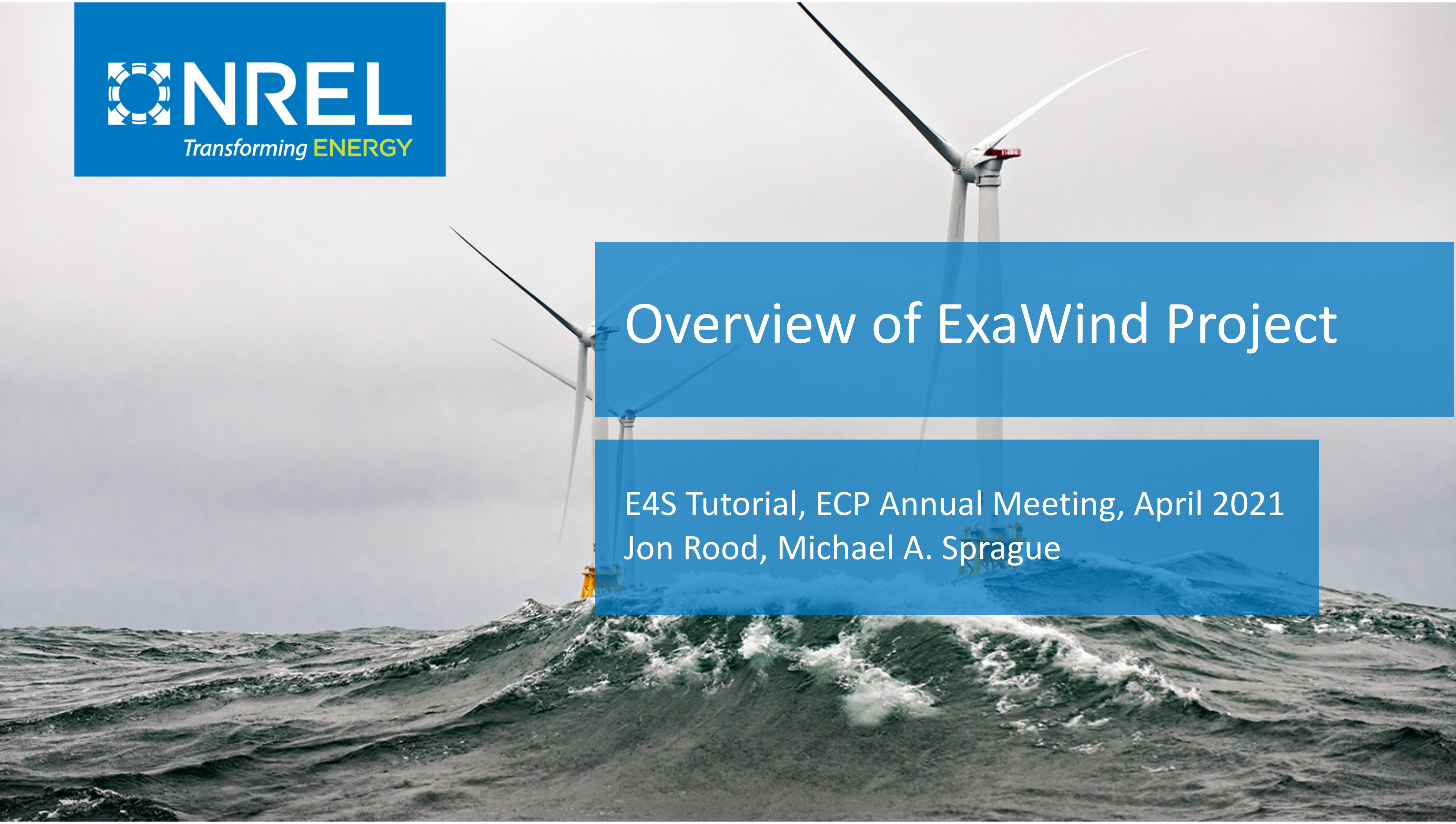
  login: tutorial

  password: HPCLinux12!

  press space bar, reenter password if necessary; open terminal window; click on README.txt

% cd; cat README

EXASCALE COMPUTING PROJECT



Overview of ExaWind Project

E4S Tutorial, ECP Annual Meeting, April 2021

Jon Rood, Michael A. Sprague

The ExaWind Project

Photo by Gitte Nyhus Lundorff, Bel Air Aviation Denmark – Helicopter Services

Helping us <u>predict</u> and <u>understand:</u>

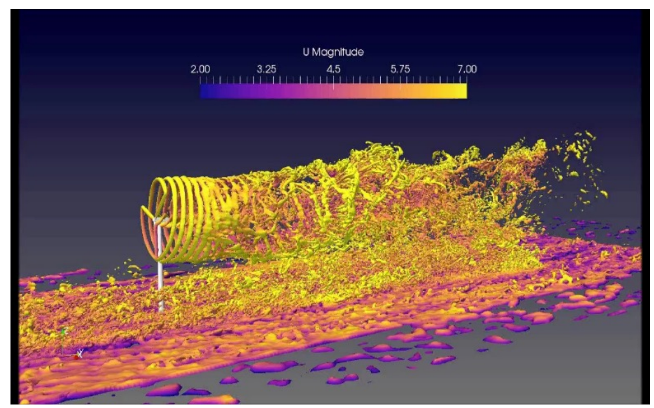
Impact of wakes on downstream turbines

Evolution of the wakes

Formation of the wakes

... and all in a highly complex, dynamic metocean environment

HFM & HPC on DOE supercomputers are keys



Team of 40 researchers

ExaWind Applications

Nalu-Wind
- Wind-specific version of the Nalu code developed at SNL
- https://github.com/exawind/nalu-wind
- Incompressible-flow computational fluid dynamics (CFD) code
- Unstructured-grid finite-volume discretization

OpenFAST
- https://github.com/openfast
- Whole-turbine simulation code
- Includes models for blades, control system, drivetrain, tower, etc.

AMR-Wind
- https://github.com/Exawind/amr-wind
- AMReX-based structured-grid CFD wind solver
- AMReX is an ECP supported library
- AMReX is also the base library for the new Energy Research & Forecasting (ERF) code

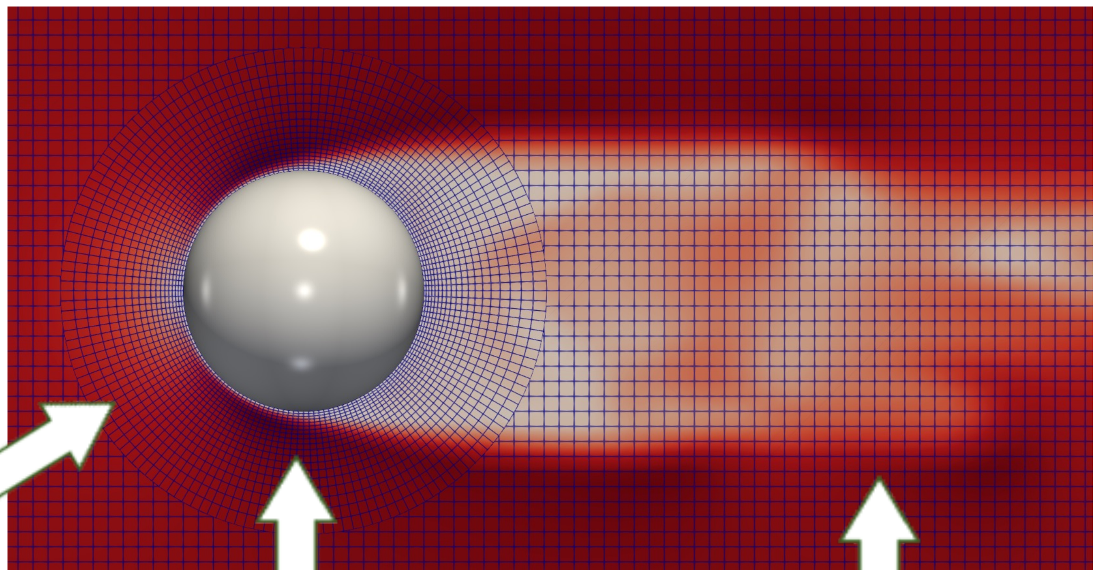
ExaWind: A multifidelity modeling and simulation environment for wind energy

M A Sprague, S Ananthan, G Vijayakumar, and M Robinson
National Renewable Energy Laboratory, Golden, CO, USA
E-mail: michael.a.sprague@nrel.gov

NAWEA 2019 paper introduced the ExaWind software stack to the wind community

Hybrid Solver Strategy

Use optimal solvers for each domain



TIOGA overset-mesh coupling

Geometry-resolving unstructured-mesh Nalu-Wind model

Background structured-mesh AMR-Wind model

Nalu-Wind/AMR-Wind flow over a sphere; proxy for wind turbine blade

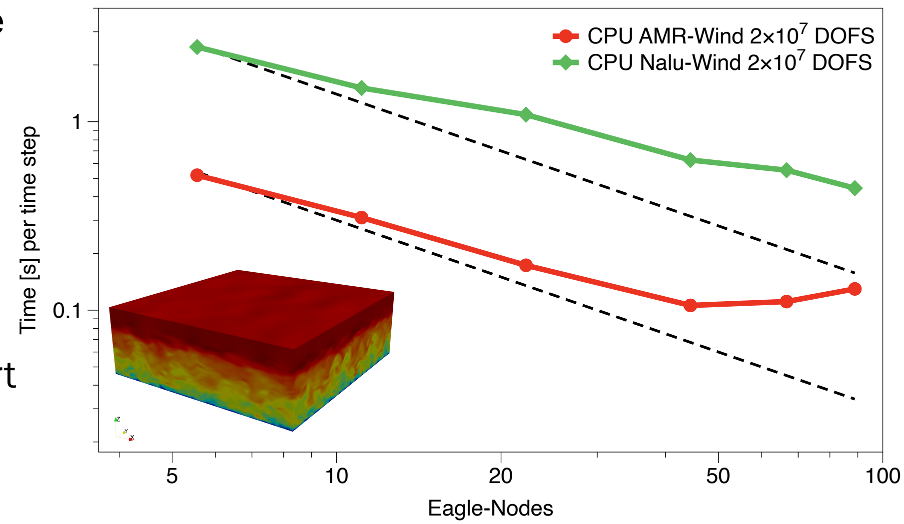
Image: Ananthan

Hybrid Solver Strategy

- In wind farm simulations, only a small fraction of the domain needs unstructured grids
  - Near turbine blades where boundary layers need to be resolved
- Vast majority of domain (atmospheric boundary layer) can be discretized with structured grids
- Created AMR-Wind, based on AMReX, to work as part of a **hybrid-solver approach** with Nalu-Wind

The hybrid-solver strategy is essential to our approach for offshore floating platform

Image: Brazell & Ananthan



Strong-scaling study on NREL Eagle HPC for large-eddy simulations of a 5 x 5 x 1 km^3 domain for a neutrally stable ABL; simulations performed on the NREL Eagle supercomputer.

- **AMR-Wind is 5x faster at scaling limit**
- **AMR-Wind simulations have reached 8.5B DOF on 15,552 GPUs on Summit**

Turbine Validation Example

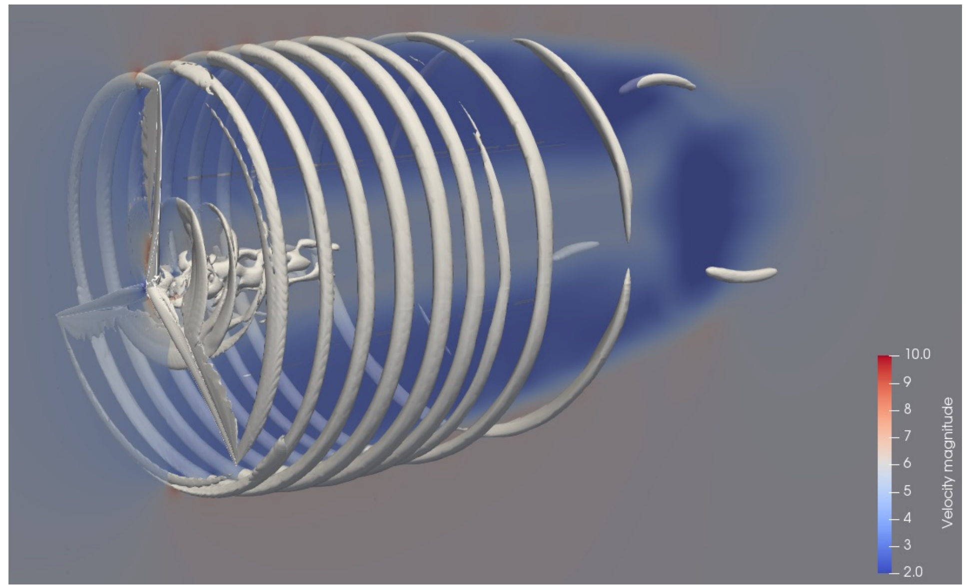
Example: NREL & DTU researchers are collaborating around the development and validation of next-generation models for predictive wind turbine simulations

The Science of Making Torque from Wind (TORQUE 2020) IOP Publishing
Journal of Physics: Conference Series **1618** (2020) 052049 doi:10.1088/1742-6596/1618/5/052049

Validation of blade-resolved computational fluid dynamics for a MW-scale turbine rotor in atmospheric flow

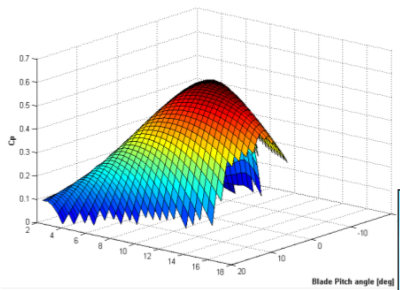
Christian Grinderslev*[1], Ganesh Vijayakumar[2], Shreyas Ananthan[3], Niels N. Sørensen[1], Frederik Zahle[1] and Michael A. Sprague[2]
[1]Department of Wind Energy, Technical University of Denmark, Risø Campus, 4000, Roskilde
[2]National Wind Technology Center, National Renewable Energy Laboratory, Golden, Colorado, USA
[3]Computational Science Center, National Renewable Energy Laboratory, Golden, Colorado, USA
E-mail: cgrinde@dtu.dk

https://iopscience.iop.org/article/10.1088/1742-6596/1618/5/052049/pdf



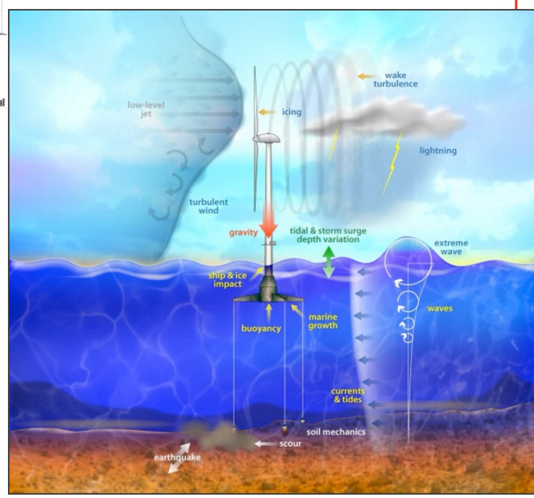
ExaWind high-fidelity computational fluid dynamics (CFD) simulation of the DanAero 2-MW wind turbine (Vijayakumar, Ananthan)

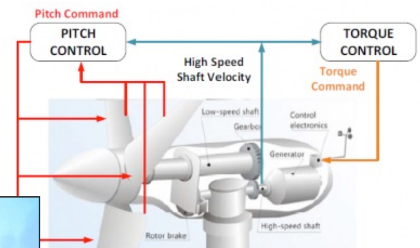
Complexities of Offshore Modeling



Aerodynamics

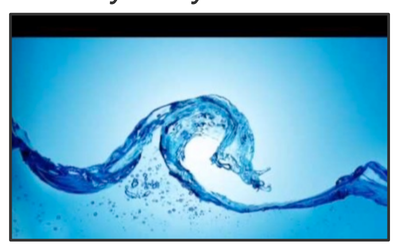
Structures





Controls

Hydrodynamics

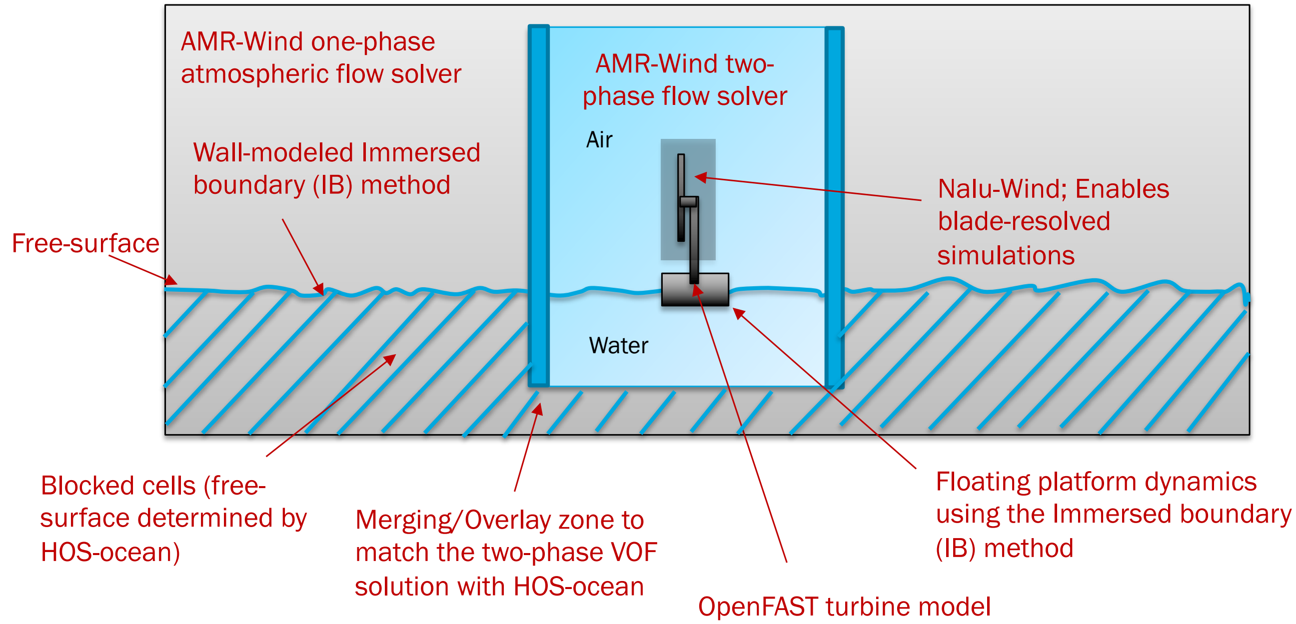


- Marine atmospheric environment (veer, shear, stability)
- Air-sea interface
- Hydrodynamics
- Wind-wave coupling
- Long-range swells and wind-wave misalignment
- Floating platform and mooring line dynamics

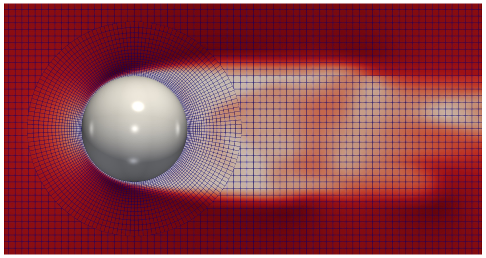
Image: J. Jonkman

EXASCALE COMPUTING PROJECT

ExaWind Vision for Offshore



CFD-Model coupling via overset, just like sphere demonstration problem



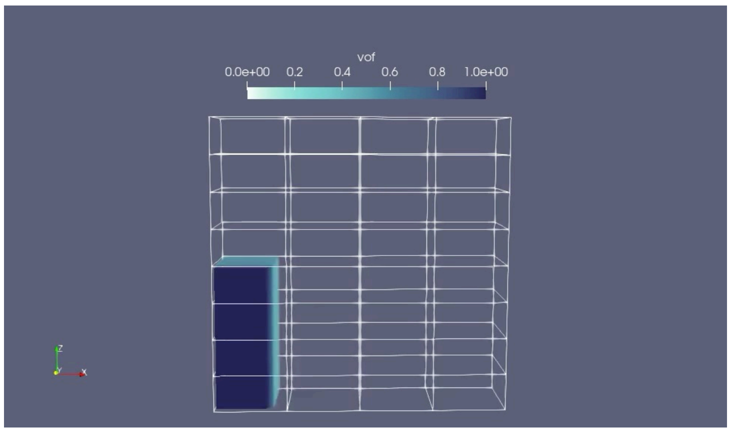
Hybrid-solver strategy is key to offshore wind:

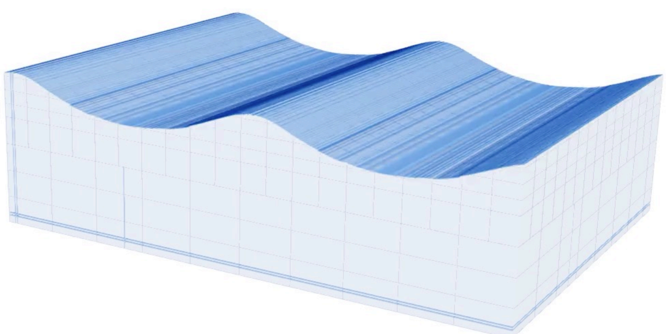
- AMR-Wind is the single-phase and two-phase background solver
  - Includes actuator lines for mid-fidelity simulations
- Nalu-Wind is the near-blade solver; coupled to AMR-Wind through overset meshes

EXASCALE COMPUTING PROJECT

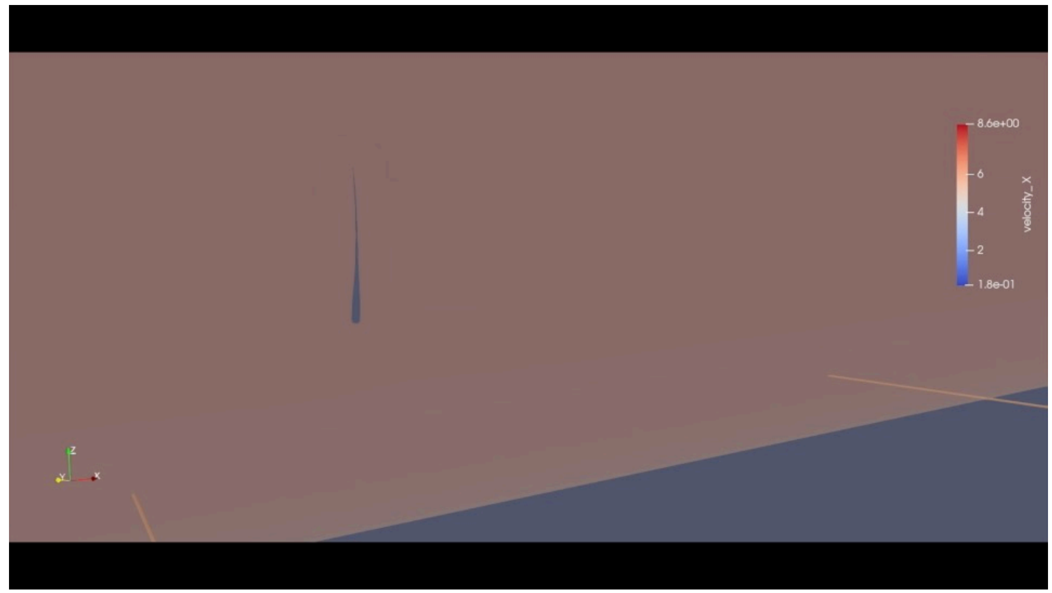
Current Capability Demonstrations

AMR-Wind volume of fluid modeling



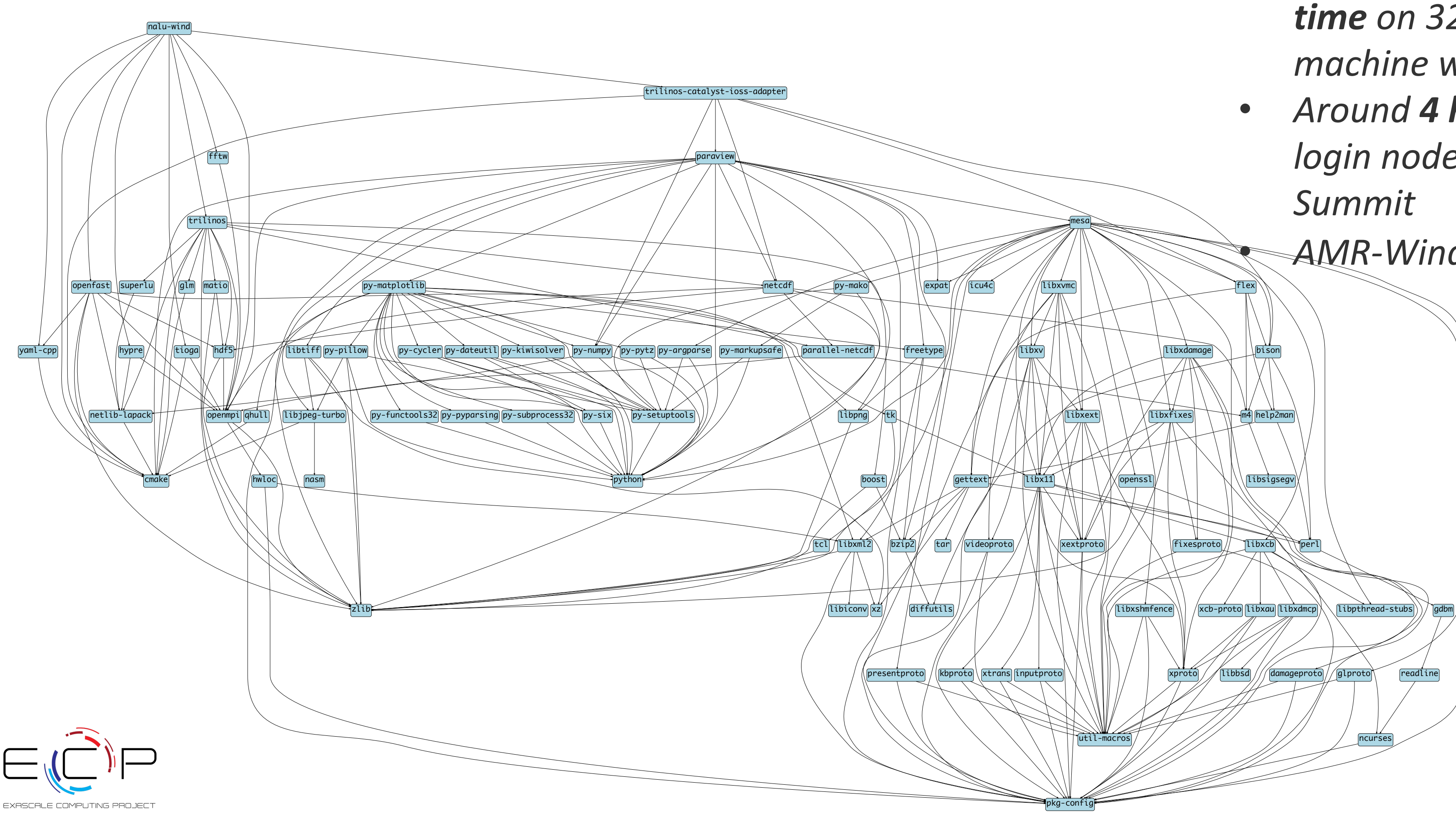


Nalu-Wind / AMR-Wind coupling



EXASCALE COMPUTING PROJECT

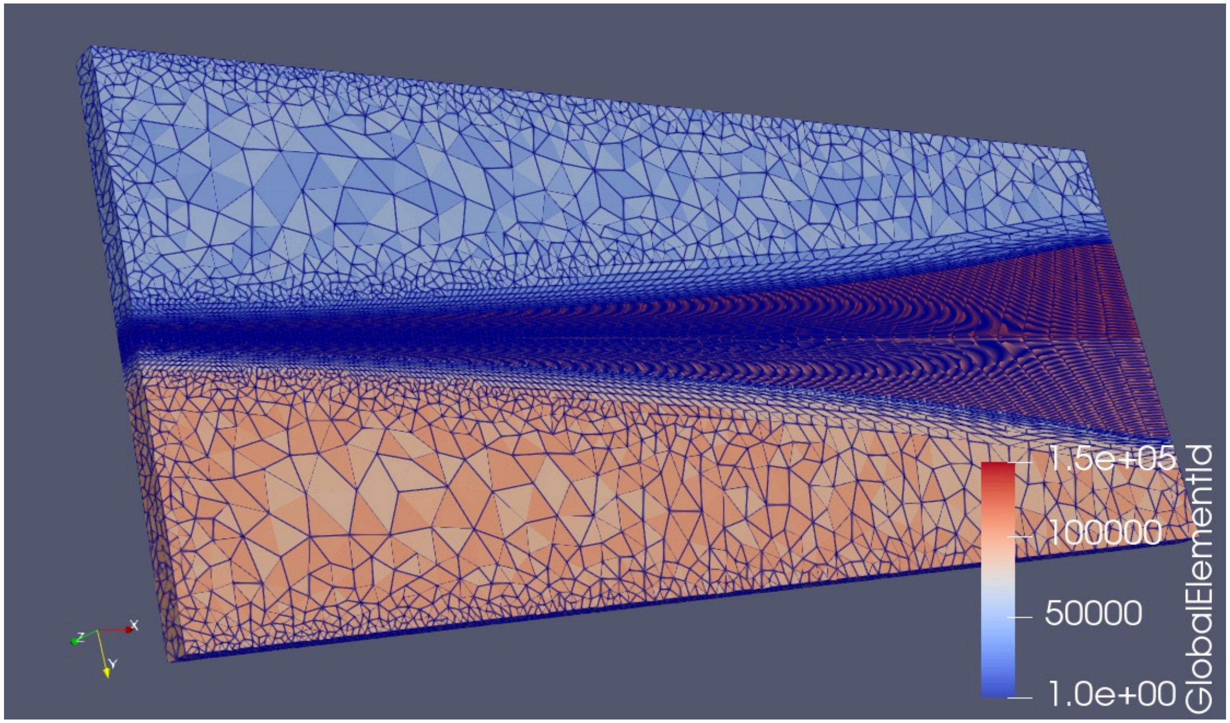
Nalu-Wind Build Graph



- *All ExaWind software in Spack*
- ***Minimum 2 hour build time** on 32-core Haswell machine with SSD*
- *Around **4 hours or more** on login node for system like Summit*
- *AMR-Wind build trivial*

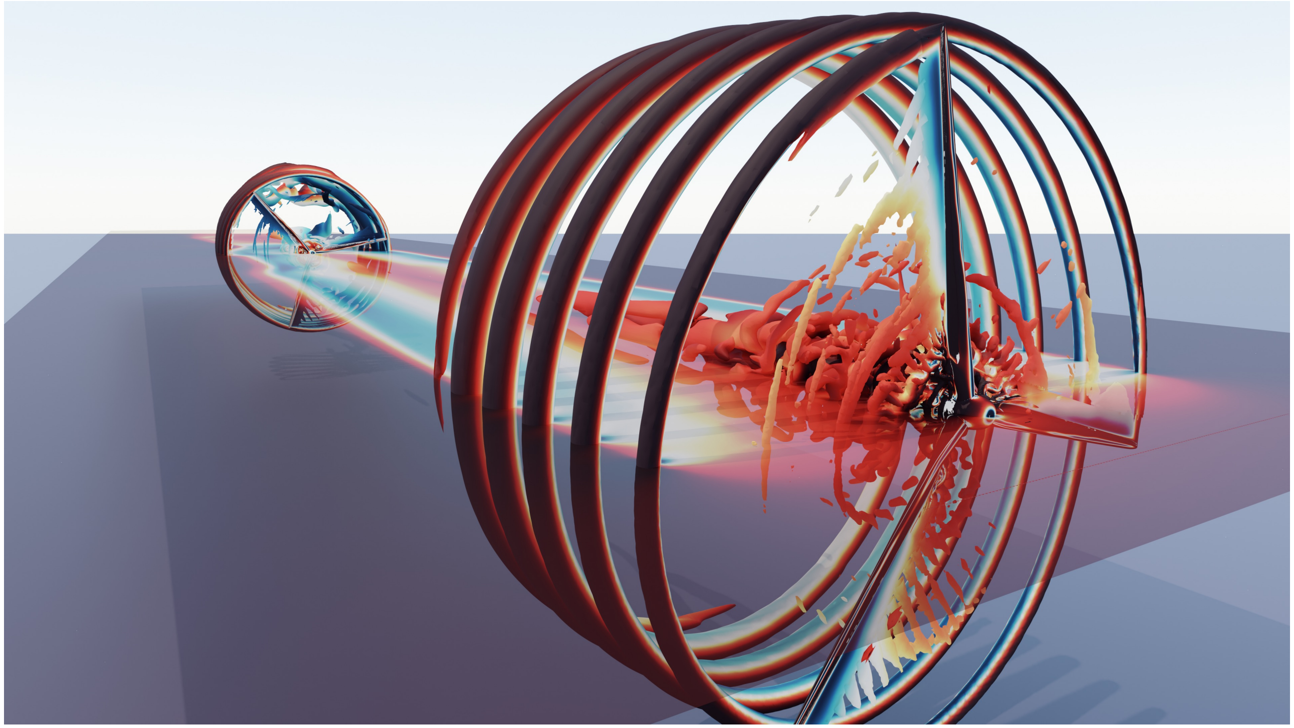
EXASCALE COMPUTING PROJECT

Description of Example in Tutorial



- Fluid mixing on odd-structured mesh
- Inflow on left edge of plate
- 90k elements
- Most visually interesting Nalu-Wind regression test

Questions



Instantaneous flow field for a blade-resolved ExaWind simulation of two NREL 5-MW wind turbines. The wind turbine tip-vortices are visualized using iso-surfaces of Q-criterion and the color contours indicate the magnitude of the velocity field on a horizontal slice passing through the hub of the two rotors.

Image credit: Ananthan, Vijayakumar, Binyahib

End

www.nrel.gov

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation's exascale computing imperative.

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the Exascale Computing Project. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government.

NREL

Transforming ENERGY